robotron
Elektronische
Rechentechnik

Systembeschreibung
K 1520

Volkmar

Volkmar

Mikrorechnersystem robotron K 1520

Inhaltsverzeichnis

1.	Einleitung
1.1.	Einsatzmöglichkeiten
2.	Systemgestaltung
2.1.	Prinzipien der Systemgestaltung
2.2.	Realisierung von Anwendungssystemen
2.3.	Technische Spezifikation der Bau- und Funktionsgruppen
2.3.1.	Zentrale Recheneinheit ZRE K 2521...K 2524
2.3.1.1.	Die Register der Zentralen Verarbeitungseinheit
2.3.1.2.	Kenndaten Zentrale Recheneinheit
2.3.2.	Halbleiterspeicher OPS, OFS, PFS
2.3.2.1.	Operativspeicher OPS K 3520
2.3.2.2.	Programmierbarer Festwertspeicher PFS K 3820
2.3.2.3.	Operativ-/Festwertspeicher OFS K 3620
2.3.3.	Bedieneinheit BDE K 7622
2.3.4.	Anschlußsteuerungen
2.3.4.1.	Anschlußsteuerung Standardinterface 1000
2.3.4.2.	Anschlußsteuerung Bedieneinheit ABD K 7022
2.3.5.	Busverstärker BVE K 4120
2.3.5.1.	Verbindungsleitung LTG K 0521 und Verbindungsleitungsadapter VLA K 0522
2.3.6.	Stromversorgungsmoduln STM
2.4.	Mikrorechnerbus K 1520
2.4.1.	Bezeichnung und Bedeutung der Signale
2.4.2.	Mehrrechnerkopplung
2.5.	Konstruktion
2.5.1.	Rückverdrahtung und Steckeinheiteneinsätze
2,2.2.	Prioritätenzuordnung
5.5.3.	Buserweiterung
2.5.4.	Steckeinheiten, Steckverbinder
2.5.5.	Aufnahmeeinrichtungen für Stromversorgungsmoduln
2.5.6.	Leitungen und Kabel für periphere Geräte
2.6.	Anschließbare periphere Geräte
2.7.	Programmierung und Systemunterlagen
2.7.1.	Befehlssatz der ZVE U 880 D
2.7.1.1.	Befehlsarten
2.7.1.2.	Adressierungsarten
2.7.1.3.	Beschreibung der Befehlsarten
2.7.2.	Basis – MOS
2.7.2.1.	K 1520 – Standardmoduln
2.7.2.2.	BASIC – Compiler K 1520
2.7.3.	Betriebssysteme
2.7.3.1.	Betriebssystem für MR-Entwicklungssystem MEOS 20
2.7.3.2.	Basisechtzeitsteuersystem BESI/BESE
2.7.4.	Wirtsrechner – MOS
3.	Sonstiges
3.1.	Umgebungsbedingungen
3.2.	Netzanschluß der Stromversorgungsmoduln K 1510
3.3.	Dimensionierung der Stromversorgung

3.4. Programmierung der EPROM
3.5. Lieferformen
3.6. Absatz
3.7. Systemkomplettierung

Anlagen

Übersicht der Gerätekomponenten Mikrorechner K 1520
Mikrorechner K 1520-Gerätekonfiguration
Blockschaltbild ZVE U 880 D
Registeraufbau ZVE U 880 D
Übersicht Systembusgenerierung
Anschluß peripherer Schaltkreise an Systembus und Koppelbus (auf K 2521)
Speicheranschluß an Systembus
Kontaktbelegung Mikrorechnerbus K 1520
Blockschaltbild OPS K 3520
Blockschaltbild OFS K 3620
Blockschaltbild PFS K 3820
Blockschaltbild Bedieneinheit
Bedien- und Anzeigefeld der Bedieneinheit
Befehlsliste ZVE U 880 D mit Erläuterungen

1. Einleitung

Durch die Anwendung der Mikrorechentechnik in der Volkswirtschaft der DDR eröffnen sich neue Möglichkeiten zur Verbesserung der Gebrauchswerteigenschaften einer Vielzahl von Erzeugnissen. Darüber hinaus sind bei der Entwicklung, Produktion und Anwendung mikrorechnergesteuerter Erzeugnisse wesentliche Rationalisierungseffekte erreichbar.
Mit dem Mikrorechnersystem robotron K 1520 stellt der VEB Kombinat Robotron eine leistungsfähige, durch modularen Aufbau und wirtschaftliche Anwendung überzeugende Neuentwicklung auf dem Gebiet der Mikrorechentechnik vor. Es ist eine folgerichtige Ergänzung der bisherigen Typen an Mikrorechnern, wozu die Modelle robotron ZE 1 und robotron K 1510 zählen.

1.1. Einsatzmöglichkeiten

Das Mikrorechnersystem robotron K 1520 ist geeignet für den vorteilhaften Einsatz auf den Gebieten:

Automatisierte Produktionssteuerung

- Steuerung und Überwachung von Prozessen innerhalb von Rechnerhierarchien
- Steuerung von Be- und Verarbeitungsmaschinen: CNC-Maschinensteuerung
- Einsatz für konventionelle industrielle Steuerungen
- Zentrale Meßwerterfassung und Überwachung von Teilprozessen
- Programmierte Maschinensteuerungen
- Prozeßsteuerungen im Fertigungsbereich

Labor- und Prüffeldautomatisierung

- Steuerung und Meßwertverarbeitung von Einzelgeräten des wissenschaftlich-technischen Gerätebaues sowie von Meß- und Prüfeinrichtungen (Integration in das Finalprodukt)
- Steurung und Meßwertverarbeitung von Gerätekomplexen, z. B. dezentrale autonome Laborautomatisierung durch Mikrolaborrechner

Informationsverarbeitungssysteme

- Steuerung von Datenein- und -ausgaben
- Steuerung von peripheren Geräten
- Sicherung von Datenübertragungen

Wissenschaftlich-technische und ökonomische Aufgaben

- Dezentrale Lösung von wissenschaftlich-technischen und technisch-ökonomischen Aufgaben mit Kleinrechnern und Terminals
- Datenerfassung in Datensammelsystemen
- Buchung, Fakturierung und Abrechnung
- Autonome dezentrale Datenerfassung und -aufbereitung

Sonstige Anwendungsgebiete

- Dezentrale Steuerung des Straßenverkehrs
- Steuerung von Verkehrsmitteln
- Steuerung medizinisch-technischer Geräte
- Überwachung und Steuerung gebäudetechnischer Anlagen
- Programmentwicklung und Testung mittels Entwicklungssystemen

Beim Einsatz des Mikrorechnersystems robotron K 1520 entstehen bedeutende Vorteile gegenüber konventionellen Ausführungen von Geräten und Anlagen.

Diese Vorteile lassen sich wie folgt charakterisieren:

- konstruktive und funktionelle Modularität und Flexibilität, d. h. gute Integrationsfähigkeit in Erzeugnisse eines Finalproduzenten, leichte Anpassung an unterschiedliche Einsatzaufgaben
- geringer Energieverbrauch
- hohe Zuverlässigkeit
- geringer Platzbedarf usw.

Um diese Vorteile voll ausnutzen zu können, sind entsprechende Kenntnisse über die Bedeutung, Einsatzmöglichkeiten und Wirkungsweise der Mikrorechentechnik notwendig. Außerdem stellt der Einsatz der Komponenten des Mikrorechnersystems K 1520 an die künftigen Anwender auch eine Reihe neuer Anforderungen, die eng mit der Einsatzzielstellung der Mikrorechentechnik verknüpft sind.
Der Entwicklungsingenieur wird in Zukunft beim Entwurf neuer Geräte und Anlagen immer prüfen, ob der Einsatz der Mikrorechentechnik der konventionellen vorzuziehen ist. Außerdem wird ein Finalproduzent unbedingt eigene Leistungen beim speziellen Anpassen und Integrieren der Mikrorechentechnik erbringen müssen.
Diese Systeminformation soll einen Überblick über das Mikrorechnersystem robotron K 1520 vermitteln und Anregungen zu dessem Einsatz geben.

2. Systemgestaltung

2.1. Prinzipien der Systemgestaltung

Gegenüber einem Kleinrechnersystem weist der Mikrorechner K 1520 einige Unterscheidungsmerkmale bezüglich der Systemgestaltung auf, die vor allem durch die speziellen Leistungsparameter der Zentralen Verarbeitungseinheit des Mikrorechnersystems und die sich daraus ergebenden anderen Einsatzzielstellungen hervorgerufen werden. Diese Unterscheidungsmerkmale drücken sich vor allem im Umfang und Komfort der Systemunterlagen, der Verarbeitungsbreite und im Spektrum der peripheren Einrichtungen aus.

Der K 1520 besteht aus einer Reihe modularer Komponenten der Gerätetechnik und Systemunterlagen. Zur Gerätetechnik gehören ein Sortiment konstruktiv und funktionell abgestimmter Bau- und Funktionsgruppen, zu denen bestückte Steckeinheiten, Steckeinheiteneinsätze, Kabel usw. zählen.

Der Einsatz der Gerätetechnik wird durch eine Reihe von Systemunterlagen unterstützt. Sie dienen der Programmentwicklung und Testung beim Anwender unter Nutzung vorhandener Rechentechnik (Wirtsrechenanlagen) sowie dem Betreiben mikrorechnergesteuerter Geräte und Anlagen. Die Systemunterlagen für den Mikrorechner bestehen aus maschinenorientierten Systemunterlagen (MOS), wozu Steuerprogrammsysteme, Ein-/Ausgabesystem zur Bedienung von Geräten bzw. Interface sowie eine bestimmte Anzahl universell verwendbarer Standardprogramme gehören. Außerdem stehen Gerätetestprogramme und Prüfsystemunterlagen zur Verfügung, die der Inbetriebnahme beim Hersteller und zur Testung des Gerätesystems beim Anwender dienen. Besonders günstig wirkt sich dabei die Zusammenfassung der Einzelprogramme für die verschiedenen Baugruppen durch ein Leitprogramm zu einem Dauertestprogramm aus.

Die Funktionseinheiten des K 1520 sind auf separaten Steckeinheiten untergebracht und können auf beliebige Steckplätze in einem zentralen Rechnerbus gesteckt werden. Durch die vorteilhafte Ausrüstung der Zentralen Recheneinheit reicht bei kleineren Leistungsanforderungen bereits eine Steckeinheit aus. Der zentrale Bus wird dabei nicht benötigt. Die logisch-funktionellen Einheiten sind in ihrer Betriebsweise für Dauerbetrieb ausgelegt. Sie sind als Steckeinheiten der Abmessungen 215 × 170 mm² ausgeführt und enthalten auf der Busseite zwei zweireihige 58polige indirekte Steckverbinder. Der Anschluß peripherer Geräte erfolgt über maximal drei 39polige indirekte Steckverbinder auf den entsprechenden Anschlußsteuereinheiten. Die Moduln (Steckeinheiten) des K 1520 sind als OEM-Einbaugruppen in Steckeinheiteneinsätzen montierbar. Dazu sind zwei Einsätze mit einer Breite von 120 mm bzw. 240 mm und gedruckter Rückverdrahtung vorgesehen.

Sind zwischen peripheren Geräten und Funktionsgruppen über Kabel Verbindungen herzustellen (z. B. zum Lochstanzer), so gehen diese von der Griffseite der dazugehörigen Anschlußsteuereinheit über die entsprechenden o. g. Steckverbinder ab.

Eine elektrische und mechanische Verlängerung des Busses ist möglich und läßt bei Bedarf eine erweiterte Konfiguration von Funktions- bzw. Baugruppen zu.

Es ist prinzipiell möglich, daß unter Einhaltung elektrischer, logischer und konstruktiver Regeln anwenderspezifische Steckeinheiten von den Anwendern des K 1520 entwickelt werden, die an den Mikrorechnerbus angeschlossen werden können. In gleicher Weise können vom Anwender des K 1520 Systemunterlagen entwickelt, angewendet und ausgetauscht werden, die über das allgemeine Angebot hinausgehen.

2.2. Realisierung von Anwendungssystemen

Durch entsprechende Auswahl einzelner Komponenten der Gerätetechnik und Systemunterlagen sowie unter Hinzunahme handelsüblicher peripherer Geräte oder spezieller kundeneigener Einrichtungen sind Anwendungssysteme realisierbar, wie sie z. B. für die im Punkt 1.1. genannten Gebiete geschaffen werden können. Unter dem Aspekt des Einsatzes in einem bestimmten Anwendungsgebiet lassen sich mit Hilfe des Sortimentes der Systemunterlagen und Systemkomponenten der Gerätetechnik:

- selbständige Mikrorechenanlagen, z B. für die Aufbereitung und Testung der Anwenderprogramme oder die Nutzung als Labormikrorechenanlage konfigurieren und
- mikrorechnergesteuerte Finalerzeugnisse, wie z. B. ein Datenerfassungsgerät oder eine CNC-Steuerung, mit diesen Komponenten ausrüsten.

Letztere Finalerzeugnisse integrieren in der Regel die Komponenten des Mikrorechnersystems vollständig, wobei sie äußerlich nicht mehr in Erscheinung treten.

2.3. Technische Spezifikation der Bau- und Funktionsgruppen

2.3.1. Zentrale Recheneinheit ZRE K 2521 ... K 2524

Die Zentrale Recheneinheit stellt das Kernstück des Mikrorechners K 1520 dar. Auf Grund des Funktionsumfanges dieser Steckeinheit ist ihr Einsatz als selbständiger Mikrorechner (Single-board-Rechner) möglich. Die Arbeitsweise der Zentralen Recheneinheit beruht auf der des Mikroprozessors U 880 D (Zentrale Verarbeitungseinheit), des Speichers (RAM, EPROM), des Zähler/Zeitgeberschaltkreises, dem parallelen Ein-/Ausgabeinterface mit Zusatzelektronik sowie quarzstabilisierten Taktgenerator und Rücksetzschaltung. Durch Abrüstung von Baugruppen werden die Varianten K 2522 (ohne Taktgenerator), K 2523 (ohne Zähler/Zeitgeberschaltkreis) und K 2524 (ohne Taktgenerator und Zähler/Zeitgeberschaltkreis) abgeleitet.

Die Zentrale Verarbeitungseinheit U 880 D bestimmt die Leistungsfähigkeit des Mikrorechnersystems K 1520 wie Verarbeitungsbreite, Wortlänge, Befehlsanzahl, Befehlsausführungsdauer usw. In der Anlage ist u. a. eine Darstellung der internen Abläufe der Zentralen Verarbeitungseinheit enthalten.

Die Zentrale Verarbeitungseinheit kann Daten aus dem Speicher oder von peripheren Geräten verarbeiten und den Datentransport mit dem Speicher und Anschlußsteuereinheiten organisieren. Dabei wird die Arbeitsweise der Zentralen Verarbeitungseinheit durch das im Speicher des K 1520 enthaltene Programm bestimmt. Der Zähler/Zeitgeberschaltkreis kann zur Echtzeitverarbeitung Unterbrechungen veranlassen und dadurch die Zentrale Verarbeitungseinheit entlasten. Über das Parallel-Ein/Ausgabeinterface können Zentrale Recheneinheiten untereinander oder externe Geräte an die Zentrale Recheneinheit gekoppelt werden.

An das Kopplungsinterface eines Masterrechners können bis zu drei weitere Slaverechner zu einem Mehrrechnersystem zusammengeschaltet werden. Der Taktgenerator erzeugt den im Mikrorechner K 1520 erforderlichen Systemtakt, und die Rücksetzschaltung bewirkt nach dem Einschalten der Stromversorgung den logischen Ausgangszustand der Zentralen Rechenenheit.

2.3.1.1. Die Register der Zentralen Verarbeitungseinheit

Der U 880 D enthält insgesamt 22 Register, davon 18 Register zu 8 Bit und 4 Register zu 16 Bit. Die Register umfassen 2 Sätze von 6 allgemeinen Registern, die als 8-Bit-Register oder paarweise als 16-Bit-Register verwendet werden können. Weitere Register sind 2 Akkumulatoren, 2 Flagregister (Statusregister), Befehlszähler, Keller-Zeiger, 2 Indexregister, Interrupt-Säulen-Register und Speicher-Auffrisch-Register. Die Operationen der Zentralen Verarbeitungseinheit laufen in einem der beiden Register/Akku-Blöcke ab, der Zugriff zum zweiten Register/Akku-Block erfolgt durch Übergangsanweisungen. Diese alternative Arbeitsweise ermöglicht wechselweises Arbeiten in Haupt- und Hintergrundprogramm ohne Auslagern von Registerinhalten in den Arbeitsspeicher und dadurch besonders schnelle Interruptbehandlung. Der 16-Bit-

Keller-Zeiger (Stack-Pointer) dient zur Bearbeitung von Mehrebenen-Interrupts, praktisch unbegrenzte Unterprogramm-Verschachtelung und Zwischenspeicherung von Datenblöcken. Die zwei 16-Bit-Indexregister erlauben die Bearbeitung von Tabellen und von im Adreßraum des Arbeitsspeichers verschiebbaren Informationen. Das Auffrischregister (Refreshregister) dient zur automatischen Speicherauffrischung der dynamischen RAM ohne Softwareaufwand, d. h., die dynamischen RAM können wie die statischen RAM benutzt werden. Das Unterbrechungsregister (Interrupt-Säulen-Register) liefert zur Realisierung einer besonders leistungsfähigen Art der Interruptbehandlung die höherwertigen 8 Bits eines Zeigers, der über eine Tabelle den Sprung in Interrupt-Behandlungsprogramme ermöglicht. In der Anlage ist der Registeraufbau der Zentralen Verarbeitungseinheit schematisch dargestellt.

2.3.1.2.
Kenndaten
Zentrale Recheneinheit

Zentrale Verarbeitungseinheit	Schaltkreistyp	U 880 D (n-Kanal-MOS)
	Verarbeitungsbreite	1 Byte (8 Bit), parallel
	Befehlslänge	1, 2, 3, 4 Byte
	Datenwortlänge	1 oder 2 Byte
	Befehlsanzahl	158 Basisbefehle
	Adressierungsraum	64 K Byte, erweiterbar
	E/A-Adreßbereich	256 Ein-/256 Ausgabeadressen, erweiterbar
	Unterbrechungsarten	– maskierte Unterbrechung, 3 verschiedene Behandlungsmodi – nichtmaskierte Unterbrechung
	Wartesteuerung	vorhanden
	Refreshsteuerung	vorhanden
	Anschlußpegel	alle Anschlüsse TTL-kompatibel
	Befehlszykluszeit	1,6 µs
	Systemtaktzyklus	407 ns ± 0,1 %
Speicher (RAM, EPROM)	Kapazität	4 K Byte, bestehend aus – 3 K Byte EPROM, abrüstbar in Stufen zu 1 K Byte – 1 K Byte statisch RAM (n-MOS)
	Adressierung	fest
	Adressen	1. EPROM $0000_H \cdots 03FF_H$ 2. EPROM $0400_H \cdots 07FF_H$ 3. EPROM $0800_H \cdots 0BFF_H$ RAM $0C00_H \cdots 0FFF_H$
Zähler/Zeitgeber	Betriebsarten	1. Zähler, 2. Zeitgeber, programmierbar
	Anzahl der Kanäle	4
	Adressierung	fest
	Adressen	$0080_H \cdots 0083_H$
	Betriebsart Zeitgeber	Anzahl der erzeugbaren Intervalle programmierbar: $(16 \cdots 256^2) \cdot t_z$, t_z = Systemtaktzykluszeit
	Betriebsart Zähler	Zählerbereich programmierbar $1 \cdots 256$ externe Ereignisse, durch Reihenschaltung von 4 Kanälen erweiterbar auf 256^4 externe Ereignisse

Parallel-Ein-/Ausgabeinterface	Anzahl der Ports	2 bidirektionale Interfacekanäle
	Adressierung	fest
	Adressen	$0084_H \cdots 0087_H$
	Betriebsarten	4
Mehrrechner-kopplungsinterface	Anzahl der koppelbaren Rechner	maximal 1 Masterrechner mit 3 Slaverechnern
	Übertragungsmodus	Interruptverkehr zwischen Zentraler Verarbeitungseinheit und E/A-Ports, programmierte Datentransfer zwischen Master- und Slaverechnern
	Übertragungs-geschwindigkeit	$\leq$ 30 K Byte/s
Anschlußbedingungen	Systembus	58polig indirekt
	Koppelbus	58polig indirekt
	Stromversorgung	K 2521 · · · K 2524: + 5 V ± 5%, max. 2,1 A — 5 V ± 5%, max. 0,1 A +12 V ± 5%, max. 0,15 A

2.3.2. Halbleiterspeicher OPS, OFS, PFS

Für die Speicheradressierung des Mikrorechners K 1520 stehen 16 Adreßbits zur Verfügung, das erlaubt, maximal 64 K Byte Speicherzellen wahlfrei zu adressieren. Die Aufrufbreite beträgt 8 Bit. Die Speicherkapazität kann je nach Erfordernis durch den Einsatzfall durch die wahlweise Ausrüstung von Festwert- und Schreib-Lese-Speichern realisiert werden. Folgende Speichermoduln stehen zur Verfügung, die beliebig und mehrfach bis zur Kapazitätsgrenze eingesetzt werden können:

OPS K 3520:	4 K Byte statischer Schreib-Lese-Speicher
PFS K 3820:	16 K Byte programmierbarer Festwertspeicher
OFS K 3620:	2 K Byte statischer Schreib-Lese-Speicher und 6 K Byte programmierbarer Festwertspeicher

Allen Speichermoduln können entsprechend dem Speichervolumen über Kodiereinrichtungen (Wickelbrücken) auf den Steckeinheiten wahlfrei zusammenhängende Adressenbereiche zugeordnet werden, wobei die Speicheranfangsadressen durch 4 K teilbare Werte annehmen. Damit ist es möglich, geschlossene Speicherfelder zu erzielen und sie den Erfordernissen der Programmsysteme anzupassen.

Die Speichersteckeinheiten werden ein- und ausgangsseitig auf den Bus parallel geschaltet. Damit ergibt sich ein steckplatzunabhängiger Einsatz der Speichersteckeinheiten.

Der Adreßbus AB 0 · · · AB 15 ermöglicht die Adressierung eines 64-K-Byte-Speichers, der nach Belieben aus OPS, OFS und PFS-Steckeinheiten zusammengestellt werden kann. Durch anwenderspezifische Ansteuerung der Signale /MEMDI 1 und /MEMDI 2 des Koppelbusses kann die Zahl der Adreßbits erweitert werden.

Die Verwendung vorgesehener Wickelbrücken (Kodierbrücken) auf den Speichersteckeinheiten für /MEMDI 1 und /MEMDI 2 ermöglicht es, die Speicherkapazität auf diese Weise einfach auf 128 K Byte zu erweitern.

2.3.2.1. Operativspeicher OPS K 3520

Der Schreib-Lese-Speicher OPS K 3520 dient zur Speicherung aller variablen Daten während des Programmablaufes im Mikrorechner K 1520.

Speicherkapazität	4 K Byte zu je 8 Bit (Matrix von 4×8 Speicherchips)
Schaltkreistyp	U 202 (1 K×1 Bit, n-MOS)
Zugriffszeit	≦ 530 ns
Betriebsarten	Lesen oder Schreiben als abgeschlossene Zyklen in beliebiger Reihenfolge
Datenerhalt	Information geht bei Abschalten der Betriebsspannung verloren. Der Datenerhalt ist prinzipiell möglich, wenn im Ruhezustand des Speichers eine gestützte Spannung von außen zugeführt wird.
Stromversorgung	+5 V ± 5%, maximal 1,6 A

2.3.2.2. Programmierbarer Festwertspeicher PFS K 3820

Der programmierbare Festwertspeicher PSF K 3820 dient der Speicherung von Festwerten für nicht variable Programme u. ä. innerhalb des Halbleiterspeichers des K 1520. Die EPROM-Schaltkreise sind über 24polige DIL-Steckfassungen auf der Steckeinheit kontaktiert. Das Beschreiben der EPROM-Schaltkreise erfolgt außerhalb der Steckeinheit auf einem sogenannten Programmierzusatz. Eine Änderung der ROM-Leseinformation ist durch Austausch oder durch Reprogrammierung der EPROM-Schaltkreise jederzeit möglich.

Speicherkapazität	16 K Byte zu je 8 Bit
Schaltkreistyp	U 555, 1 K×8 Bit (n-MOS)
Zugriffszeit	≦ 530 ns
Betriebsarten	Lesen als abgeschlossener Zyklus
Datenerhalt	Energieunabhängige Speicherung von Festdaten
Stromversorgung	+ 5 V ± 5%, maximal 0,9 A — 5 V ± 5%, maximal 0,5 A +12 V ± 5%, maximal 0,9 A

2.3.2.3. Operativ-/Festwertspeicher OFS K 3620

Der kombinierte Schreib-Lese-Speicher und programmierbare Festwertspeicher OFS K 3620 dient der Speicherung von variablen sowie Festdaten innerhalb des Halbleiterspeichers K 1520. Dieser Speichermodul ermöglicht in der Kombination mit den anderen Speichermoduln eine Flexibilität in der Zusammensetzung der Halbleiterspeicher und realisiert ökonomische kleine Speicher. Für die RAM- und EPROM-Schaltkreise gelten die gleichen Angaben wie unter den Ausführungen zu den OPS und PFS.

Speicherkapazität	2 K Byte zu je 8 Bit statisch RAM 6 K Byte zu je 8 Bit EPROM
Schaltkreistypen	U 202 (RAM), U 555 (EPROM)
Zugriffszeit	≦ 530 ns
Betriebsarten	Abgeschlossene Zyklen Lesen oder Schreiben in beliebiger Reihenfolge beim RAM und Lesen beim EPROM
Datenerhalt	Wie bei OPS und PFS
Stromversorgung	+ 5 V ± 5%, maximal 1,2 A — 5 V ± 5%, maximal 0,2 A +12 V ± 5%, maximal 0,3 A

2.3.3. Bedieneinheit K 7622

Die Bedieneinheit BDE K 7622 dient zur Unterstützung der Kommunikation mit dem Mikrorechner K 1520 bei der Inbetriebnahme, Wartung und Fehlersuche sowie bei der Testung von Programmen. Der Anschluß der Bedieneinheit an den Mikrorechnerbus erfolgt mittels eines 3 m langen Flachbandkabels über die Steckeinheit Anschlußsteuerung Bedieneinheit ABD K 7022. Die Bedieneinheit wird als Auftischgerät mit Frontplattenbedienung aufgebaut. In der Anlage ist u. a. eine Ansicht der Frontplatte der Bedieneinheit enthalten. Die Bedieneinheit ist nicht Bestandteil der Grundausstattung einer Mikrorechnerkonfiguration!

Abmessungen	525 mm × 240 mm × 240 mm
Betriebsspannung	220 V/50 Hz (siehe auch 3.2.)
Stromverbrauch	0,13 A
Schutzgrad	IP 20
Einsatzklasse	EKL 5/40/30/95 10-1_E
Steuerung	25 Leuchtdrucktasten rastend 7 Leuchtdrucktasten nichtrastend
Anschluß an K 1520-Bus	3 m Flachbandkabel (Zubehör BDE) über ABD K 7022
Anzeigen	11 Steuersignale (LED) Datenbus 8 Bit dual (LED) Adreßbus 16 Bit hexadezimal (LED)

Beschreibung

Die Bedieneinheit ermöglicht:

- das Starten von beliebigen Adressen
- das Stoppen zu beliebigen Zeiten
- das Stoppen auf frei wählbare Adressen (sogenannten Testpunkten)
- das Arbeiten des Mikrorechners im Einzelzyklusbetrieb
- die Bereitstellung von Adressen, Daten und Steuersignalen für den Mikrorechner
- den direkten Speicherzugriff (für statische und dynamische Speicher)
- die Anzeige von Adressen, Daten und Zuständen der Zentralen Recheneinheit des K 1520

Funktion

Registerstruktur:

Die Bedieneinheit enthält ein Adreßregister für 16 Adressenbits. Der Adreßregisterinhalt wird hexadezimal über die Luminiszensanzeigen VQB 71 dargestellt. Mit dem Tastenschalter „LAD ADR" wird die mit den Tastenschaltern 0···15 eingestellte Adresse in das Register geladen.

Das Datenregister der Bedieneinheit umfaßt 8 Bit. Die Anzeige von gültigen Daten erfolgt über die den Leuchtdrucktastern 0···7 zugeordneten 8 LED's. Das Einschreiben der Daten in das Register erfolgt mit dem Tastenschalter „LAD DAT".

HALT durch die Bedieneinheit:

Bei Betätigen des Tastenschalters „HALT" wird unter der Bedingung, daß es sich um einen Befehlslesezyklus handelt, der 1-Byte-Steuergruppenbefehl „HALT" erzeugt. Über die Bus-Treiber werden die Daten an die Zentrale Recheneinheit gegeben.

START durch die Bedieneinheit:

Beim Betätigen des Tastenschalters „START ADR" wird das Signal /NMI gesetzt und anschließend der Rücksprungbefehl RETN erzeugt. Die Signale RETN M1 und RETN M2 senden den 2-Byte-Rücksprungbefehl aus. Über die Signale RETN M3 und RETN M4 wird die im Adreßregister stehende Startadresse ausgegeben.

Testpunkt:

Zum Erkennen des Testpunktes wird der Inhalt des Adreßregisters mit der Stellung der Tastenschalter 0···15 verglichen. Bei Übereinstimmung der eingetasteten und der vom Bus empfangenen Adresse wird das Signal HALT OPCO erzeugt und die Zentrale Recheneinheit in den HALT-Zustand gebracht.

Speicher schreiben:

Nach Anliegen der Tastersignale „ERL WR", „WR" und dem Quittungssignal /BUS AK von der Zentralen Recheneinheit werden taktgerecht die Signale /MREQO und WRO gebildet. Die in das Adreßregister und das Datenregister eingegebenen Signale werden mit Betätigung des Tastenschalters „START ADR" in den Speicher eingeschrieben. Nach jeder Schreiboperation erfolgt ein automatisches Lesen der eingestellten Adresse mit Vergleich der eingestellten Daten und der vom Speicher gelesenen. Bei Übereinstimmung leuchtet die Anzeige „DATOK" ca. 1 s auf.

Speicher lesen:

Bei Betätigen der Tastenschalter „BUSRQ" und „RD" werden die Signale /MREQO und RDO nach Erhalt des Quittungssignales /BUSAK gebildet. Über die eingestellte Adresse werden die dazugehörigen Daten bei Drücken der Taste „START ADR" ausgelesen.

Refreshsteuerung:

Die Bedieneinheit sendet nach Drücken der Taste „BUSRQ" oder „ZYKL" Refreshadressen aus, um bei Einsatz von dynamischen Speichern und längerer Busherrschaft den Datenerhalt zu gewährleisten. Die Rückgabe der Busherrschaft an die Zentrale Recheneinheit erfolgt erst nach einer kompletten Auffrischperiode.

Zyklischer Betrieb:

Der Tastenschalter „ZYKL" erzeugt das Signal WAIT, welches die Zentrale Recheneinheit in den Wartezustand versetzt. Dabei wird von der Bedieneinheit die Adresse mit den dazugehörigen Daten angezeigt. Bei Betätigen des Tastenschalters „START ZYKL" wird der nächste Zyklus angelaufen.

Nichtmaskierter Interrupt:

Der Tastenschalter „NMI" aktiviert das Signal NMI, welches die Zentrale Recheneinheit zu einem RESTART ab Speicherplatz 0066_H zwingt.

2.3.4.1. Anschlußsteuerung Standardinterface daro 1000

Die Anschlußsteuereinheiten ADA K 6020 und K 6021 dienen zum Anschluß von peripheren Geräten mit dem Standard interface daro 1000/1 an den Mikrorechner K 1520. Der Anschluß erfolgt mittels Peripheriekabels (siehe 2.5.6.). An die 3 Ein-/Ausgabekanäle der Steckeinheiten können 2 Ausgabegeräte und ein Eingabegerät (ADA K 6020) bzw. 1 Ausgabegerät und 2 Eingabegeräte (ADA K 6021) angeschlossen werden. Die Steckeinheiten können unter Beachtung der Prioritäten (siehe 2.5.2.) steckplatzunabhängig an den Systembus angeschlossen werden. Der Datenaustausch erfolgt grundsätzlich interruptgesteuert.

Übertragungsgeschwindigkeit	20 K Byte/s, 8 Bit parallel ohne Paritätsbit 5 K Byte/s, 8 Bit parallel mit Paritätsbit
Anzahl der Kanäle	K 6020: 2 Ausgabekanäle, 1 Eingabekanal K 6021: 1 Ausgabekanal, 2 Eingabekanäle
Übertragungsentfernung	maximal 20 m
Signalpegel	KME 3, H-Potential +6,5···12 V L-Potential 0···+0,5 V
Geräteanschluß	Standardanschluß daro 1000/1, realisiert durch 3×39polige Steckverbinder nach TGL 29 331/04
Belastung der Ausgangsleistungen	maximal 3 mA bei +12 V
Stromversorgung	K 6020 und K 6021: + 5 V ± 5%, maximal 1,1, A +12 V ± 5%, maximal 0,25 A

2.3.4.2. Anschlußsteuerung Bedieneinheit ABD K 7022

Die Anschlußsteuerung Bedieneinheit ist nur in der Zusammenschaltung mit der Bedieneinheit K 7622 aktiv. Sie ist auf die spezifischen Bedingungen der Bedieneinheit abgestimmt und realisiert die Weiterleitung der anzuzeigenden Busspannungen, bzw. die Einspeisung der durch Tastenfunktionen der Bedieneinheit manipulierten Bussignale auf den Systembus. Die Steckeinheit ist wegen des standardisierten Anschlußbildes des Systembusses prinzipiell steckplatzunabhängig einsetzbar, die Plazierung muß aber so erfolgen, daß diese Steckeinheit die erste Stelle der Prioritätskette für Busanforderungen einnimmt (siehe 2.5.2.). Die Bedieneinheit wird mittels einer Flachbandleitung an die griffseitige Buchsenleiste der Anschlußsteuerung angekoppelt.

Stromversorgung	+5 V ± 5%, maximal 0,9 A

2.3.5. Busverstärker BVE K 4120

Der Busverstärker ermöglicht in der Kopplung über die Verbindungsleitungen LTG K 0521 (1 Paar) und den Verbindungsleitungsadapter VLA K 0522 eine Verlängerung des System- und Koppelbusses des Mikrorechners K 1520, wenn die in einem Steckeinheiteneinsatz zur Verfügung stehende Anzahl von Steckeinheitenplätzen für die gewünschte Rechnerkonfiguration nicht ausreicht und größere Entfernungen zwischen den zu koppelnden Steckeinheiteneinsätzen überbrückt werden müssen. Der Busverstärker muß in dem Steckeinheiteneinsatz, der die Zentrale Recheneinheit enthält, untergebracht werden. Funktionell besteht keine Begrenzung in der Anzahl der zur Busverlängerung gekoppelten Steckeinheiteneinsätze. Zu beachten sind dabei lediglich die Belastungs- und Zeitverhältnisse auf dem System- und Koppelbus des Mikrorechners K 1520.

Stromversorgung	+5 V ± 5%, maximal 1,0 A

2.3.5.1.
Verbindungsleitung LTG K 0521 und Adapter VLA K 0522

Der Verbindungsleitungsadapter und 2 Stück Verbindungsleitungen gehören zum notwendigen Zubehör des Busverstärkers BVE K 4120. Die Leitungslänge der Verbindungsleitungen kann bis maximal 2,5 m betragen.

2.3.6.
Stromversorgungsmoduln STM

Zur Stromversorgung des Mikrorechners K 1520 können u. a. die Stromversorgungsmoduln des Mikrorechnersystems K 1510 eingesetzt werden. Diese Stromversorgungsmoduln in den Leistungsklassen 20 W und 50 W wandeln die Netzspannung, 220 V Einphasenwechselspannung, in entsprechende potentialfreie Gleichspannungen um. Die Stromversorgungsmoduln arbeiten autonom und können in einem dafür vorgesehenen Baugruppeneinsatz aus dem Lieferumfang des Mikrorechnersystems K 1510 steckbar untergebracht werden. Ein Netzfiltermodul NFI K 0311 vermindert sämtliche im Mikrorechner entstehenden Störspannungen (Funkstörgrad F 1 nach TGL 20 855/05). Die Stromversorgungsmoduln und der Netzfilter besitzen EGS-Einbaumaße von 135 mm × 170 mm × 30 · · · 55 mm. Alle Stromversorgungsleitungen müssen über Schraubverbindungen an den Modulen befestigt werden.

Im folgenden wird eine kurze Übersicht über die für die Stromversorgung des Mikrorechners K 1520 geeigneten Stromversorgungsmoduln gegeben. Eine umfassende Übersicht und Beschreibung aller im Sortiment des Mikrorechnersystems K 1510 enthaltenen Stromversorgungsmoduln kann der Systembeschreibung des Mikrorechnersystems K 1510 entnommen werden.

STM K 0310.01	12 V/1,5 A
STM K 0310.05	5 V/3,3 A
STM K 0310.06	5 V/10 A
STM K 0310.09	12 V/4,2 A
NFI K 0311	

2.4.
Mikrorechnerbus K 1520

Der Rechnerbus des Mikrorechners K 1520 wird durch 2 Bündel Signalleitungen sowie die Leitungen zur Stromversorgung der Rechnerbaugruppen gebildet, die konstruktiv zwei 58poligen Steckverbindern jeder Steckeinheit zugeordnet sind.

Das erste Leitungsbündel umfaßt die zum Systemaufbau unbedingt notwendigen Hauptsignale und wird als Systembus bezeichnet; er bildet die gemeinsame Verbindung zwischen der Zentralen Recheneinheit, den Speichern, den E/A-Einheiten und sonstigen Funktionsgruppen zur Durchführung des Informationsaustausches zwischen diesen. Konstruktiv ist er im Steckeinheiteneinsatz durch eine gedruckte Rückverdrahtung in Form durchgehender Leitungen realisiert.

Das zweite Leitungsbündel umfaßt Signale, die, abhängig von der Ausrüstung der Zentralen Recheneinheit, vorhanden sind und die Kopplung mehrerer Rechner zu einem Mehrrechnersystem ermöglichen, sowie die Zeittakt- und Steuersignale der Echtzeituhr und Stromversorgungs- und Überwachungsleitungen. Der zur Mehrrechnerkopplung benutzte Signalbündelteil wird als Koppelbus bezeichnet. Seine freien Steckverbinderanschlüsse können in der Rückverdrahtung zur Herstellung spezifischer Verbindungen zwischen den Steckverbindern benutzt werden.

In der Anlage wird in einer anschaulichen Darstellung die Systembusgenerierung durch die Zentrale Recheneinheit des Mikrorechners K 1520 demonstriert, in einer weiteren dazugehörigen Tabelle ist die Signalbelegung des Systembusses und des Koppelbusses an den Wickelstiften enthalten.

Der Systembus des Mikrorechners K 1520 wird durch folgende Leitungsgruppen gebildet:

Leitungsgruppe	Anzahl (PIN)	Bezeichnung
Datenbus	8	DB 0···DB 7
Adreßbus	16	AB 0···AB 15
Steuer- und Kontrollbus	20	/MREQ, /IORQ, /M1, /RFSH, /IODI, /BUSRQ, /BUSAK, /MEMDI, /RD, /WR, /WAIT, /RDY, /HALT, /RESET, /NMI, /INT, /IEI, /IEO, /BAI, /BAO, Takt
Stromversorgung	14	5 P, 12 P, 5 N, 5 PG, 00

2.4.1. Bezeichnung und Bedeutung der Signale

Die einzelnen Signale besitzen folgende Bedeutung:
DB 0···DB 7 (Datenbus)
Die Leitungen führen beim Datenaustausch auf dem Bus die Befehls- bzw. Dateninformation

AB 0···AB 15 (Adreßbus)
Die Leitungen führen die Adresse für Informationsaustausch mit dem Speicher bzw. E/A-Gerät. AB 0···AB 15 sind mit /MREQ als Speicheradresse gültig (≙ 64 K Byte).
AB 0···AB 7 sind mit /IORQ als E/A-Geräteadressen gültig (256 Eingabe-/256 Ausgabeadressen).
AB 8···AB 15 sind mit /IORQ gültig und enthalten bei Ein- und Ausgabebefehlen den Inhalt des ZVE-Registers A bzw. B (je nach Befehlsart) bzw. den Inhalt des oberen Byteregisters des DMA-Kanals.
AB 0···AB 6 sind mit /RFSH als Refreshadresse für das Auffrischen dynamischer RAM's gültig.

/MREQ (Speichertransfergesuch)
Das Signal zeigt an, daß der Adreßbus eine gültige Adresse für eine Speicherlese- bzw. -schreiboperation führt.

/IORQ (Ein-/Ausgabetransfergesuch)
Das Signal zeigt an, daß der Adreßbus im unteren Byte (AB 0···AB 7) eine gültige E/A-Geräteadresse führt und die Daten auf AB 8···AB 15 gültig sind. Das Signal wird von der ZRE auch erzeugt, wenn ein Interruptgesuch von der ZVE akzeptiert wurde und der neue Befehl bzw. Interruptvektor auf den Datenbus gelegt werden darf (siehe /M1).

/RD (Lesen)
Das Signal zeigt an, daß durch die ZVE bzw. den DMA-Kanal Informationen (Daten oder Befehle) vom Speicher bzw. E/A-Gerät gelesen werden sollen.

/WR (Schreiben)
Das Signal zeigt an, daß von der ZVE bzw. den DMA-Kanal gültige Daten auf den Datenbus gelegt wurden, die im Speicher einzutragen bzw. vom E/A-Gerät zu übernehmen sind.

/RFSH (Auffrischen)
Das Signal zeigt an, daß die unteren 7 Bit des Adreßbusses eine Refreshadresse zum Auffrischen dynamischer RAM's bilden. In Verbindung mit /MREQ sollte dieses Signal benutzt werden, jeweils eine Refresh-Leseoperation durchzuführen.

/M1 (Befehlslesezyklus)
Das Signal zeigt an, daß der laufende ZVE-Zyklus ein Befehlslesezyklus des auszuführenden Befehls ist. In Verbindung mit dem Signal /IORQ zeigt es an, daß ein Interruptgesuch akzeptiert wurde und der Interruptvektor auf den Datenbus zu legen ist.

/BUSRQ (Gesuch auf direkten Speicherverkehr)
Das Signal zeigt der ZVE an, daß ein Gesuch auf direkten Speicherverkehr durch einen DMA-Kanal gestellt wurde. Das Signal besitzt bezüglich der ZVE-Unterbrechung höchste Priorität.

/BUSAK (Anerkennung des direkten Speicherverkehrs)
Mit diesem Signal zeigt die ZVE an, daß das Signal /BUSRQ akzeptiert wurde und Datenbus, Adreßbus und 3-state Steuerausgänge in den hochohmigen Zustand geschaltet sind. Der Bus steht dem DMA-Kanal zur Arbeit zur Verfügung.

/WAIT (Warten)
Das Signal zeigt an, daß Speicher bzw. E/A-Gerät nicht für einen Datenaustausch „bereit" sind. Die ZVE tritt so lange in einen Wartezustand ein, wie das Signal aktiv ist.

/RDY (Bereit)
Das Signal zeigt auf dem Bus an, daß der angesprochene Speicher oder das angesprochene E/A-Gerät am Bus vorhanden ist und für Lese- oder Schreiboperationen zur Verfügung steht.

/RESET (Rücksetzen)
Das Signal zeigt an, daß ein zentrales Rücksetzen in der ZVE erfolgt. In der ZVE bewirkt es folgende Reaktionen:

- Rücksetzen des internen Interrupt-FF
- Löschen des Befehlszählers
- Löschen Interrupt-Vektor-Register
- Löschen Refresh-Adreß-Register
- Setzen Interrupt-Mode 0

Bei Einschalten der Betriebsspannung S P erzeugt die Zentrale Recheneinheit automatisch ein RESET-Signal auf dem Systembus (Nennwert 500 ms).

/INT (Maskiertes Unterbrechungsgesuch)
Das Signal kennzeichnet ein Unterbrechungsgesuch eines peripheren Gerätes an die ZVE. Bei /BUSRQ = aktiv wird diese Leitung bei DMA-Betrieb auch unter der Bedingung „Datenblockende" durch den DMA-Kanal aktiviert.

/NMI (Nichtmaskiertes Unterbrechungsgesuch)
Das Signal kennzeichnet ein Unterbrechungsgesuch an die ZVE. Es besitzt höhere Priorität als /INT und wird unabhängig vom Zustand des ZVE-internen Interrupt-Enable-FF akzeptiert. /NMI zwingt die ZVE, bei der Adresse 0066H zu starten.
Diese Leitung sollte vorzugsweise für Netzausfallinterrupt benutzt werden.

/IEI (Interrupt-Freigabe-Eingang)
Das Signal kennzeichnet, daß die am Bus näher zur Zentralen Recheneinheit gesteckte Steckeinheit keinen Interrupt anfordert und die Priorität der nachgeschalteten Steckeinheit zugewiesen ist.

/IEO (Interrupt-Freigabe-Ausgang)
Das Signal kennzeichnet, daß die Steckeinheit kein Interrupt fordert. Die Leitung ist direkt mit /IEI der nachfolgenden Steckeinheit zu verbinden.

/MEMDI (Speichersperrung)
Das Signal dient der Sperrung aller Speichermodule für Lese- und Schreiboperationen. Die Signale /MREQ und /M1 werden davon nicht beeinflußt. Das Signal wird bei Simulation des Speichers durch die Bedieneinheit bzw. einen externen Speicher benutzt.

Signal geht zur CPU
Falls die CPU IORQ bildet,
wird im Normalfall RDY
zurückgesetzt
IODI verhindert das

/IODI (E/A-Gerätesperrung)

Das Signal dient der Sperrung des E/A-Systems und verhindert die automatische Eingabe von Interruptadressen bzw. -befehlen. Die Signale /IORQ und /M1 werden davon nicht beeinflußt. Das Signal wird bei der Simulation des E/A-Systems durch die Bedieneinheit bzw. E/A-Simulatoren und für die Handeingabe von Interruptadressen bzw. -Befehlen verwendet.

TAKT (Systemtakt)

Die zugeordnete Leitung führt den zentralen Systemtakt. Er entspricht dem ZVE-Grundtakt und steht den Busteilnehmern zur zeitlichen Synchronisation interner Vorgänge zur Verfügung.

/BAI und /BAO (Anerkennung des direkten Speicherverkehrs – Eingang/Ausgang)

Die Signale BAI/BAO bilden am Bus eine prioritätsbestimmende Kette zur Durchschaltung des Signals /BUSAK. Dabei bildet BAI das Eingangssignal (BUSAK-Input) und BAO (BUSAK-Output) das weitergeleitete Anerkennungssignal am Ausgang der Steckeinheit.

/HALT (ZVE-Halt)

Das Signal zeigt an, daß sich der Halt-Zustand der ZVE eingestellt hat. Die ZVE wartet nun zur Weiterarbeit auf ein Interrupt.

2.4.2. Mehrrechnerkopplung

Die Kopplung mehrerer Rechnerkonfigurationen K 1520 erfolgt durch die Verbindung der entsprechenden Anschlüsse des Koppelbusses in direkter Verdrahtung. Falls diese Verbindungen nicht innerhalb der Rückverdrahtung eines 480 mm breiten EGS-Baugruppeneinsatzes bleiben, sind die sich aus der Leitungslänge ergebenden Zusatzmaßnahmen, z. B. Einschaltung von Verstärkern, vom Anwender spezifisch zu realisieren.

Es sind insgesamt 4 Rechnerkonfigurationen koppelbar: 1 Masterrechner und 3 Slaverechner.

Jede Rechnerkonfiguration kann für sich ein vollständiger Mikrorechner K 1520 mit an einem Systembus angeschlossenen Speichersteckeinheiten, Anschlußsteuerungen usw. sein.

2.5. Konstruktion

2.5.1. Rückverdrahtung und Steckeinheiteneinsätze

Die Steckeinheitenmoduln des Mikrorechners K 1520 können in die Steckeinheiteneinsätze STS K 0120 und STS K 0121 eingeschoben werden. Der Steckeinheiteneinsatz STS K 0120 ist 120 mm breit und bietet 5 Steckeinheiten Platz. Der Steckeinheiteneinsatz STS K 0121 ist 240 mm breit und bietet 11 Steckeinheiten Platz.

Sowohl komplette Steckeinheiteneinsätze als auch einzelne Steckeinheiten sind beliebig kombiniert einsetzbar in spezifische Gefäße der Anwender, z. B. in die 480 mm breiten EGS-Baugruppeneinsätze.

Bei den Steckeinheiteneinsätzen sind sowohl die Steckverbinderanschlüsse der Rückverdrahtung des Koppelbusses als auch die des Systembusses mit langen Wickelstiften versehen, so daß auf den gedruckten Rückverdrahtungsleiterplatten beliebige Verbindungen zusätzlich gewickelt werden können. Beim Systembus (unter Rückverdrahtungsleiterplatte) besteht im allgemeinen keine Notwendigkeit von Zusatzbewicklungen, da alle Signalanschlüsse durch die gedruckten Leiterzüge miteinander verbunden sind.

Beim Koppelbus sind nur die Masse- und Stromversorgungsanschlüsse gedruckt ausgeführt. Alle weiteren Verbindungen sind zu wickeln. Der Koppelbus ist somit speziell auch für die Verbindung der Anschlüsse anwenderspezifischer Steckeinheiten geeignet.

Bei Verwendung von spezifischen Gefäßen durch den Anwender ist durch letzteren auch die Rückverdrahtung der Steckeinheiteneinsätze zu realisieren, in die die Rückverdrahtung der Steckeinheiteneinsätze integriert sein kann. Es können mehrere Steckeinheiteneinsätze zusammengeschraubt und die Busleitungen durch Wickelverbindungen zwischen den benachbart liegenden Randsteckverbinderanschlüssen rückverdrahtungsseitig geführt werden. Auf diese Weise können Einschubgefäße für die Aufnahme von 16, 22 usw. Steckeinheiten geschaffen werden.

2.5.2. Prioritätenzuordnung

Jede Steckeinheit ist zunächst an eine beliebige Stelle im Steckeinheiteneinsatz steckbar. Jedoch bestimmt die Entfernung, in der eine Steckeinheit mit peripheren Schaltkreisen, d. h. zum Anschluß peripherer Geräte, von der Zentralen Recheneinheit ZRE K 2521 bis K 2524 gesteckt wird, die Prioritäten dieser Steckeinheit bei der Bedienung von Interruptanforderungen (IEI-IEO-daisy-chain-Kette) oder bei der Anforderung der Busherrschaft (BAI-BAO-daisy-chain-Kette). Je geringer der Abstand von der Zentralen Recheneinheit ist, um so höher ist ihre Priorität.

Die Speichersteckeinheiten (OPS, OFS und PFS) zählen dabei nicht mit, da auf ihnen die Daisy-chain-Ketten gebrückt sind. Sind mehrere periphere Schaltkreise auf einer Steckeinheit, dann ist die interne Reihenschaltung der Prioritäts-Ein-/Ausgänge zu beachten. Z. B. auf der Zentralen Recheneinheit ZRE K 2521 hat der Zähler/Zeitgeber-Schaltkreis die höhere Interruptpriorität gegenüber dem Parallel-Ein-/Ausgabe-Schaltkreis. Diese beiden peripheren Schaltkreise haben, da sie sich direkt auf der Zentralen Recheneinheit befinden, natürlich eine höhere Interruptpriorität als alle anderen.

Die Anzahl der in der Interrupt-Prioritäts-Kette hintereinander schaltbaren peripheren Bausteine hängt von deren spezifischen Daten ab.

Nicht belegte Steckplätze sind entweder mit Brücken IEI-IEO und BAI-BAO (Kurzschlußstecker oder in Rückverdrahtung gewickelt) zu versehen, oder sie sind alle bündig anzuordnen.

2.5.3. Busweiterleitung

Der Systembus des Mikrorechners K 1520 ist in der Rückverdrahtung bis zu einer Länge von 800 mm aufbaubar, wobei Stichleitungen an den Steckeinheiten kleiner 100 mm lang sein sollten. Durch Zwischenschaltung eines Busverstärkers BVE K 4120 kann der Systembus belastungsmäßig um maximal 3 m verlängert werden. Das Verlängerungskabel ist mittels Steckverbinders in den griffseitigen Anschluß des Busverstärkers zu führen, der in der konstruktiven Einheit des 1. Busses steckt. Die andere Seite des Kabels ist über eine Adapter-Steckeinheit VLA K 0522 auf einen Steckplatz des Gefäßes (z. B. Steckeinheiteneinsatz STS K 0121), das den 2. Bus realisiert, zu stecken.

Der 2. Bus erhält als ganzes die Priorität des Steckplatzes des Busverstärkers im 1. Bus. Damit die Prioritätenkette im 2. Bus geschlossen wird, sind die Anschlüsse IEO und BAO des letzten besetzten Steckplatzes (oder des letzten Steckplatzes im Gefäß, wenn an den freien Steckplätzen analog 2.5.2. die Brücken IEI-IEO und BAI-BAO realisiert werden) zurück an die Anschlüsse IEI und IEO des ersten (besetzten) Steckplatzes dieses Busses (2.) zu führen, z. B. mittels einer gewickelten Drahtbrücke.

Nachdem die Prioritätskette des 2. Busses durchlaufen ist, pflanzt sich der Prioritätendurchlauf über IEO bzw. BAO des Steckplatzes des Busverstärkers im 1. Bus fort.

Beim Datentransfer mit der ZVE können die Laufzeiten durch den Aufbau eines Shake-Hand-Spiels mit den Bussignalen /RDY und /WAIT berücksichtigt werden, was dem Anwender obliegt. Unter diesen Bedingungen ist auch ein sternförmiges (mehrere Busverstärker in einem Bus) oder serielles (je Bus eine Busverstärkereinheit) System mit mehr als 2 Bussen aufbaubar.

2.5.4. Steckeinheiten, Steckverbinder

Alle Funktionsgruppen des Mikrorechners K 1520 sind auf jeweils einer einzigen Steckeinheit untergebracht. Einzige Ausnahme bildet die Funktionseinheit Foliespeicheranschluß, die u. a. in der Ergänzung zu dieser Systembeschreibung vorgestellt wird.

Alle Steckeinheiten sind Zweiebenenleiterplatten vom Format 215 mm × 170 mm. Zum Anschluß an den System- und Koppelbus werden indirekte Steckverbinder (PdAu), auf jeder Steckeinheit jeweils eine für den entsprechenden Bus mit der Bezeichnung 304-58 TGL 29 331/03 und den dazugehörigen Buchsenleisten in der Rückverdrahtungsleiterplatte 434-58 TGL 29 331/03, eingesetzt. Der Busverstärker besitzt zusätzlich griffseitig auf der Steckeinheit 2 Stück Buchsenleisten 202-58 TGL 29 331/03 und am dazugehörigen Verbindungskabel LTG K 0521 je eine Steckerleiste 102-58 TGL 29 331/03.

Die beiden Anschlußsteuereinheiten ADA K 6020, K 6021 besitzen griffseitig auf der Sieckeinheit zum Anschluß von Peripheriekabeln jeweils 3 Stück Buchsenleisten 402-39 TGL 29331/04. Sollte eine Nutzung der im Lieferumfang des Mikrorechnersystems K 1510 enthaltenen Peripheriekabel nicht möglich sein, sind entsprechend selbst anzufertigende Kabel mit der Steckerleiste 322-39 TGL 29 331/04 auszurüsten.

2.5.5. Aufnahmeeinrichtungen für Stromversorgungsmoduln

Für die Stromversorgungsmoduln aus dem Mikrorechnersystem K 1510 können auch die dazugehörigen Aufnahmeeinrichtungen aus dem Lieferumfang der OEM-Baugruppen des K 1510 eingesetzt werden. In diesen Aufnahmeeinrichtungen, die in den Breiten 480 mm oder 440 mm (zum Einbau in 19-Zoll-Gefäße) angeboten werden, können je nach Typ der Stromversorgungsmoduln bis maximal 7 Stück Stromversorgungsmodule und Netzfilter aufgenommen werden. Genaue Angaben können der Betriebsdokumentation des Mikrorechnersystems K 1510 entnommen werden.

Kurzbez.	Chiffre	Breite	Höhe	Tiefe	Verw. zur Aufn. von
BGS	K 0117	440 mm	160 mm	240 mm	STM, NFI
BGS	K 0118	480 mm	160 mm	240 mm	STM, NFI

2.5.6. Leitungen und Kabel für periphere Geräte

Zum System- und Lieferumfang des Mikrorechnersystems K 1520 gehören keine Kabel zum Anschluß der peripheren Geräte (analog Stromversorgung). Es wird empfohlen, im Falle des Bedarfes auf das Liefersortiment Kabel aus dem Mikrorechnersystem K 1510 zurückzugreifen. Eine vollständige Übersicht über die Kabel des Mikrorechnersystems K 1510 ist in der Betriebsdokumentation K 1510 enthalten. Im folgenden sind nur die wichtigsten aufgeführt:

Bezeichnung	Länge	Kurzbez.	Chiffre
Kabel für Lochbandleser daro 1210	5 m 10 m 15 m 20 m	KAB KAB KAB KAB	K 0514.01 K 0514.02 K 0514.03 K 0514.04
Kabel für Lochbandstanzer daro 1215 und SD 1156	5 m 10 m 15 m 20 m	KAB KAB KAB KAB	K 0515.01 K 0515.02 K 0515.03 K 0515.04
Spezialkabel für SD 1156		KAB	K 0515.05
Kabel für V 24-Anschluß	5 m 10 m 15 m	KAB KAB KAB	K 0517.01 K 0517.02 K 0517.03

2.6. Anschließbare periphere Geräte

Prinzipiell können über die Anschlußsteuereinheiten ADA K 6020, ADA K 6021 alle Geräte, die das Standardinterface 1000/1 im KME 3-Pegel realisieren, angeschlossen werden. Das sind im einzelnen z. Z. folgende Geräte:

- Lochbandleser 1210-0333 oder 1210-0332 mit Elektronikkassette
- Lochbandstanzer mit Elektronikkassette 1215-1011 und Netzteil 1060-1001
- Bediendrucker 4000
- Seriendrucker SD 1156/ASC II
- Kassettenmagnetbandgerät 1250-1 mit Elektronikkassette 1250-22

Die Angaben über die KME 3–Pegelgrenzen sind dem Punkt 2.3.4.1. zu entnehmen. Im Rahmen der Systemkomplettierung ist der Anschluß weiterer peripherer Geräte vorgesehen. In der Systembeschreibung Teil II werden die Anschlußsteuereinheiten für Folienspeicher, Bildschirmanzeigebaugruppe, Tastatur und den Drucker 1157 vorgestellt.

2.7. Programmierung und Systemunterlagen

2.7.1. Befehlssatz der ZVE U880D

2.7.1.1. Befehlsarten

Die Befehle der ZVE U 880 D lassen sich in 8 Gruppen oder Befehlsarten einordnen.

- Die Lade- und Austauschbefehle ermöglichen den Datentransport zwischen den internen Registern der ZVE bzw. zwischen diesen und externen Speichern. Diese Befehle enthalten die Angabe der entsprechenden Ziel- und Quellregister bzw. -speicher. Die Austauschbefehle dienen dem Vertauschen der Inhalte zweier Internregister.
- Zum Speichern, Lesen und Suchen von Datenblöcken variabler Länge im Speicher stehen Blockübertragungs- und Blocksuchbefehle zur Verfügung. Diese Befehle führen beim Erfüllen vorgegebener Parameter zum automatischen Abbruch,

lassen sich aber auch unterbrechen, um die ZVE nicht für lange Zeitabschnitte zu blokkieren.

- Die arithmetischen und logischen Befehle verknüpfen den Inhalt des Akkumulators mit dem der internen Register oder der externen Speicher. Dabei werden die vorhandenen Flags in Abhängigkeit vom Resultat beeinflußt.
- Der Registerinhalt und der externe Speicher können durch Umlauf- und Verschiebebefehle verändert werden.
- Die Bitbehandlungsbefehle ermöglichen das Setzen, Löschen und Testen eines beliebigen Bits in den Registern oder im Speicher.
- Für Programmverzweigungen und die Benutzung von Unterprogrammen stehen Sprung-, UP-Aufruf- und Rücksprungbefehle zur Verfügung. Diese Befehle beeinflussen den Inhalt des Befehlszählers.
- Der Datentransport zwischen den peripheren Geräten und den Internregistern bzw. Speicherplätzen wird durch E/A-Befehle erreicht. Dabei ist es möglich, daß verschiedene E/A-Tore die allgemeine Treiberroutine benutzen können, indem durch einen speziellen Befehl das C-Register als Toradresse definiert wird und als Zeiger für die E/A-Geräteadressierung verwendet werden kann. Es gibt bei der ZVE U 880 D Befehle zum Transport von Datenblöcken mit einer maximalen Länge von 256 Bytes zwischen einem beliebigen Speicherbereich und dem angewählten peripheren Gerät in beide Richtungen.
- Die ZVE-Steuerbefehle ermöglichen das Setzen der Interruptanforderungs-Flip-Flops und definieren den jeweiligen Behandlungsmodus.

2.7.1.2. Adressierungsarten

Die durch die ZVE zu verarbeitenden Daten stehen entweder als Direktoperanden, als Inhalt von Internregistern, Speicherplätzen oder E/A-Toren zur Verfügung. Sie müssen also adressiert werden. Das gilt auch für Sprungbefehle, bei denen die Adresse die Stelle der Programmfortsetzung anzeigt.

Direkte Adressierung

Der Operand ist das dem Operationscode folgende Byte.

Beispiel: Laden des Akkumulators mit einem Wert.

Erweiterte direkte Adressierung

Die 2 dem Operationscode folgenden Bytes bilden den Operanden, wobei das erste Byte den niederen Teil und das zweite den höheren Teil beinhaltet.

Beispiel: Transport einer 16-Bit-Adresse in das Registerpaar HL.

Modifizierte A-Byte-Adressierung

Die ZVE U 880 D besitzt spezielle UP-Aufrufe, bei denen der Operationscode (1 Byte) die Adresse (3 Bits) enthält. Hiermit lassen sich 8 Speicherplätze im Anfangsbereich des Speichers adressieren. Häufig aufzurufende Unterprogramme können so platzsparend adressiert werden.

Beispiel: Restart-Befehl

RST 0 = C7 RST 20 = E7
RST 8 = CF RST 28 = EF
RST 10 = D7 RST 30 = F7
RST 18 = DF RST 38 = FF

Relative Adressierung

Das dem Operationscode folgende Byte enthält die relative Entfernung in Adreßeinheiten, zu der ein Sprung ausgeführt werden soll. Diese im Zweierkomplement dargestellte Entfernung wird zur Adresse des folgenden Operationscodes addiert.
Diese Adressierungsart ermöglicht mit einem 2-Byte-Befehl den Sprung zu naheliegenden Speicherplätzen (relativer Sprung). Da bei vielen Programmen die hiermit erreichbare Adressenverschiebung von +129 bis —126 gegenüber der Bezugsadresse ausreichend ist, ergeben sich beträchtliche Speicherplatzeinsparungen.

Erweiterte Adressierung

Zur Kennzeichnung eines beliebigen Speicherplatzes sind 2 Bytes erforderlich. Das erste Byte nach dem Operationscode beinhaltet den niederen, das zweite den höheren Teil der Adresse.

Beispiel: Sprungadresse, Datentransport von oder zu einem beliebigen Speicherplatz.

Indizierte Adressierung

Das dem in diesem Fall aus 2 Byte bestehenden Operationscode folgende Byte enthält die Anzahl Adreßeinheiten, die zum Inhalt eines der beiden Indexregister (im Operationscode definiert) addiert wird und als Resultat die tatsächliche Adresse ergibt. Der Inhalt des Indexregisters bleibt unverändert. Die Verschiebung wird im Zweierkomplement dargestellt.

Beispiel: Arbeit mit Datentabellen, wobei das Indexregister den Anfang der Tabelle angibt.
Als Schreibweise wird (IX+d) oder (IY+d) gewählt.

Registeradressierung

Der Operationscode enthält die Bezeichnung des Ziel- und Quellregisters.

Beispiel: Laden des Akkumulators mit dem Inhalt des C-Registers.

Unbedingte Adressierung

Der Operationscode enthält automatisch die Adresse eines oder mehrerer ZVE-Register. So steht zum Beispiel das Resultat einer arithmetischen Operation immer im Akkumulator.

Indirekte Registeradressierung

Die 16-Bit-Adresse eines Registerpaares (z. B. HL) im OP-Code verschlüsselt wurde
Inhalt eines Registerpaares (z. B. HL) im 1- oder 2-Byte-Operationscode angegeben.

Bitadressierung

Drei Bits im Operationscode kennzeichnen, welches der 8 Bits eines Registers oder Speicherplatzes verarbeitet werden soll.

2.7.1.3. Beschreibung der Befehlsarten

Die Befehlsliste der ZVE U 880 D ist als Anlage in dieser Systembeschreibung enthalten. Beim Programmieren wird die Assembler-Mnemonik geschrieben. Die Codierung erfolgt in hexadezimaler Darstellung. In diesem Abschnitt sollen einige Erläuterungen zu den bereits genannten Befehlsarten gegeben werden.

Laden und Austauschen

Diese Befehle bewirken den Transport von Daten in Register, zwischen den Registern und zwischen Registern und Speicherplätzen. Dem Operationscode folgen stets Ziel und Quelle (immer in dieser Reihenfolge!). Dabei können verschiedene Adressierungsarten auftreten. Die Mnemonik der Operation lautet LD.

Ladebefehle, die die indizierte Adressierung zur Kennzeichnung der Quelle oder des Zieles verwenden, sind 3-Byte-Befehle, wobei im dritten Byte die Verschiebung zum Inhalt des Indexregisters angegeben wird.

Beispiel: Der dritte Wert einer Tabelle, deren Anfangsadresse minus 1 in IY steht, soll in den Akkumulator geladen werden.

Mnemonik:	LD A, (IY+3)	
Codefolge:	A : FD	) OP-Code
	A+1: 7E	)
	A+2: 03	Verschiebung

Ladebefehle mit erweiterter Adressierung benötigen ebenfalls 3 Byte.

Beispiel: Der Inhalt des Speicherplatzes 3333H soll in den Akkumulator übernommen werden.

Mnemonik:	LD A, (3333H)	
Code:	A :3A	OP-Code
	A+1:33	niederer) Adreßteil
	A+2:33	höherer)

Das direkte Laden der 8-Bit-Register mit einem Wert erfordert einen 2-Byte-Befehl.

Beispiel: Der Wert 42H soll in das Register E geschrieben werden.

Mnemonik:	LD E, 42H
Code:	A :1E
	A+1:42

Bei indizierter Adressierung eines Speicherplatzes, der mit einem Direktoperanden geladen werden soll, sind 4 Byte erforderlich.

Beispiel: Der Wert 36H soll in den Speicher IY-15 geladen werden.

Mnemonik:	LD (IY-15), 36H	
Code:	A :FD	)
	A+1:36	) OP-Code
	A+2:F1	
	A+3:36	

Zwischen den Registern und den Speicherplätzen sind 16-Bit-Transportoperationen möglich. Dabei umfaßt die erweiterte Adressierung alle Registerpaare.

Mit Hilfe der PUSH- und POP-Befehle ist es möglich, einen beliebigen Teil des Speichers als Kellerspeicher zu verwenden. Die PUSH-Befehle bewirken, daß ein Stapelregisterzeiger (Stackpointer) automatisch um 1 vermindert wird. Dabei wird der höherwertige Teil eines Doppelwortes gekellert. Nach einer weiteren Dekrementierung des Stackpointers folgt das Abspeichern des niederwertigen Teils.

Der POP-Befehl bildet die Umkehrung des PUSH-Befehls. Der niederwertige Teil wird gelesen. Es folgen die Erhöhung des Stackpointers um 1 und das Lesen des höherwertigen Teils aus dem definierten Stapel.

Man beachte also, daß stets der höherwertige Teil zuerst gekellert und zuletzt gelesen wird.
Besonders vorteilhaft sind die Austauschbefehle „EX" und „EXX" der ZVE U 880 D.

Sie ermöglichen den Austausch der Inhalte von Registerpaaren, den Transport von Doppelworten aus dem Keller in das Registerpaar HL und die Indexregister sowie den Inhaltsaustausch zwischen den beiden Registergruppen (siehe Registeraufbau in der Anlage). Diese Befehle erlauben den einfachen Aufbau einer schnellen Interruptbehandlung.

Blocküberträgungs- und Blocksuchbefehle

Diese Gruppe beinhaltet sehr leistungsstarke Befehle. Diese arbeiten stets mit 3 Registerpaaren, die vom Programmierer zu füllen sind.

Dabei beinhaltet	HL die Adresse der Quelle,
	DE die Adresse des Zieles.
	BC bildet einen Bytezähler.

So wird durch den Befehl LDI ein Byte vom Speicherplatz, dessen Adresse in HL steht, zum in DE adressierten Speicherplatz transportiert. Danach werden beide Registerpaare um 1 erhöht und gleichzeitig der Inhalt des Registerpaares BC um 1 erniedrigt. Beim LDIR-Befehl wird dieser Vorgang so lange wiederholt, bis der Inhalt von BC (Bytezähler) gleich Null ist. Mit einem Befehl kann also ein Datenblock von einem in einen anderen Speicherbereich umgelagert werden. Dabei kann die Blocklänge maximal 64 K Byte betragen. Eine Registerpaarüberlappung ist möglich.

Eine Umkehrung dieser Befehle stellen die LDD- und LDDR-Befehle dar. Sie erlauben die Blocküberträgung vom Ende eines Blockes aus, d. h. der Inhalt der Adreßregister HL und DE wird jeweils dekrementiert.

Das Durchsuchen des Speichers nach einem Byte erfolgt mit Hilfe der Blocksuchbefehle. Beim CPI-Befehl wird der Inhalt des Akkumulators mit dem des Speicherplatzes, dessen Adresse in HL steht, verglichen. Das Ergebnis wird im Flag-Register gespeichert. HL wird erhöht und der Bytezähler (BC) erniedrigt. Beim CPIR-Befehl wird dieser Vorgang so lange wiederholt, bis die verglichenen Werte gleich sind oder der Bytezähler gleich Null ist.

Das analoge Durchsuchen des Speichers in umgekehrter Richtung erfolgt durch die CPD- und CPDR-Befehle, wobei der Inhalt von HL nach jedem Vergleich erniedrigt wird.

Arithmetische und logische Befehle

Die arithmetischen 8-Bit-Operationen bewirken (außer bei Inkrementierung und Dekrementierung) Verbindungen zwischen dem Akkumulator und adressierten Quelldaten. Außer beim Vergleichsbefehl steht das Ergebnis im Akkumulator. In Abhängigkeit vom Ergebnis werden die Flags beeinflußt. Bei indizierter Adressierung folgt die Angabe der Verschiebung direkt nach dem Operationscode. Die Befehle ADD und SUB realisieren eine binäre Addition bzw. Subtraktion im Akkumulator. Bei den Befehlen ADC und SBC wird das Übertragsflag in die Operation einbezogen. Mit Hilfe der Flags und einer Dezimalanpassung (DAA) können arithmetische Operationen ausgeführt werden für:

- Mehrfachgenauigkeit gepackter BCD-Zahlen
- Mehrfachgenauigkeit markierter oder unmarkierter Zahlen
- Mehrfachgenauigkeit des Zweierkomplements markierter Zahlen.

Die in der Befehlsliste im Anhang genannten 5 arithmetischen Mehrzweckbefehle (DAA bis SCF) arbeiten mit dem Carry-Flag. Sie dienen der Vereinfachung des Aufbaus einer Dezimalarithmetik. Ein Löschbefehl für das Carry-Flag existiert nicht, da das Löschen durch ein logisches UND des Akkumulatorinhalts mit sich selbst realisiert werden kann.

Die logischen Operationen AND, OR, XOR und CP (Vergleichen) gehören ebenfalls zu dieser Befehlsgruppe.

16-Bit-Operationen beziehen sich auf die Registerpaare BC, DE, HL, SP, IX, IY, wobei bei ADC und SBC das Carry-Flag einbezogen wird.

Umlauf- und Verschiebebefehle

Diese Befehlsgruppe erlaubt den Umlauf einzelner Bits im Akkumulator, in jedem Register und auf beliebigen Speicherplätzen. Dabei wird das C-Flag einbezogen. Ferner sind auch einfache Verschiebungen nach links und rechts möglich. Zur Erleichterung der Arbeit mit Dezimalzahlen (BCD-Zeichen) gibt es Befehle, die den Umlauf zwischen 1 Zeichen im Akkumulator (Bit b_0–b_3) und 2 Zeichen des in (HL) adressierten Speicherplatzes ermöglichen. Dies wird für den Aufbau einer leistungsstarken BCD-Arithmetik genutzt.

Bitbehandlungsbefehle

Diese Befehlsgruppe umfaßt 240 verfügbare Befehle. Dabei sind Registeradressierung, indirekte Adressierung und indizierte möglich. Jedes Bit eines so adressierten Registers oder Speicherplatzes kann gesetzt, gelöscht oder getestet werden. Falls beim Test das betreffende Bit gleich Null ist, wird das Z-Flag gesetzt.

Sprung-, Ruf- und Rückkehrbefehle

Sprungbefehle dienen der Verzweigung von Programmen. Es sind direkte, relative und indirekte Adressierung möglich. Bei ersterer folgt dem Operationscode unmittelbar der niedere Teil der Adresse, danach der höhere.

Zwei Bytes genügen zur Darstellung des Sprungbefehls mit Relativadresse, da das 2. Byte die Verschiebung (im Zweierkomplement) zum aktuellen Befehlszähler darstellt (von 129 bis –126). Indirekte Sprünge lassen sich durch Voreinstellung der Register HL, IX und IY realisieren.

Beim Rufbefehl (CALL) wird die Adresse des ihm folgenden Bytes in den Keller (Stack) geladen. Dieser Befehl wird zum Ansprung von Unterprogrammen verwendet.

Der Rückkehrbefehl (RET) übernimmt die Adresse von der Spitze des Stacks direkt in den Befehlszähler und führt danach den Rücksprung aus.

Die ZVE U 880 D besitzt spezielle Rücksprungbefehle, die als unbedingte Rücksprünge behandelt werden:

RETI Rücksprung vom maskierbaren Interrupt,

RETN Rücksprung vom nichtmaskierbaren Interrupt.

Der Befehl RETI wird als Endbefehl einer Interrupt-Routine peripherer U 880D-Chips verwendet zur Steuerung eigener verschachtelter Prioritäts-Interrupt-Behandlungen. Der 2-Byte-Befehl RETI ersetzt dabei die 7-Byte-Folge Interrupt verbieten, Mitteilung an Peripherie, daß Routine beendet, Interrupt erlauben und Rücksprung.

Sehr vorteilhaft ist der Befehl DJNZ, der die Steuerung von Programmschleifen erleichtert. Dabei wird das vorher einzustellende B-Register bei relativer Sprungadressierung dekrementiert. Solange sein Inhalt $\pm$ 0 ist, wird der Sprung ausgeführt. Es ist also keine eigene Zählung der Durchläufe erforderlich.

Ein- und Ausgabebefehle

Bei dieser Befehlsgruppe erfolgt die Adressierung des jeweiligen E/A-Gerätes entweder absolut oder durch den Inhalt des C-Registers indirekt. Bei den verfügbaren 8 Blockübertragungsbefehlen (analog der Blockübertragung im Speicher) wird das Registerpaar HL als Zeiger für die Quelle (Ausgabe) bzw. das Ziel (Eingabe) benutzt, während das B-Register als Bytezähler fungiert. Das C-Register enthält dabei die Toradresse des peripheren Gerätes. Die maximale Blocklänge beträgt 256 Bytes.

Bei den Transporten zwischen Peripherie und Akku werden die Geräteadressen in der unteren Hälfte des Adreßbusses (A_0–A_7) und der Wert in der oberen (A_8–A_{15}) übertragen.

Bei den indirekten E/A-Befehlen (auch Blockübertragung) wird der Inhalt des Registers C in der unteren Hälfte als Geräteadresse und der des B-Registers in der oberen des Adreßbusses übertragen.

Realisiert wird die Ein- und Ausgabe über einen programmierbaren Parallel-I/O-Interfacebaustein (PIO). Dieser ist u. a. gekennzeichnet durch:

- Quittungsbetrieb im Interruptmode,
- 4 Betriebsarten pro Tor: Byte-Ausgabe,
  Byte-Eingabe,
  Byte-E/A (bidirektional nur bei Tor A),
  Bit-E/A,
- programmierbare Interruptbehandlung unter Bedingungen des peripheren Gerätes,
- automatische Interruptvektorerzeugung und Prioritätscodierung durch Kaskadierung der Bausteine.

Vor der ersten Interruptanforderung wird der jeweils gewünschte Vektor von der ZVE in den PIO als Steuerwort an das betreffende Tor geschrieben. Die Auswahl der Betriebsart erfolgt in den beiden höchsten Bits eines weiteren Steuerwortes. Bei der Bit-E/A ist ein zusätzliches Steuerwort erforderlich.

Die Interruptsteuerung erfolgt durch ein eigenes Steuerwort (Verbot bzw. Erlauben eines Interrupts).

ZVE- Steuerbefehle

Diese 7 Befehle steuern nur die ZVE. Dabei stellt der Befehl NOP einen Leerbefehl dar, während der HALT-Befehl die ZVE stoppt bis zum Empfang eines nachfolgenden Interrupts.

Zwei Befehle (DI, EI) verbieten oder erlauben einen Interrupt. Drei weitere Befehle bestimmen den Interrupt-Antwortmodus der ZVE.

Bei nichtmaskierbarem Interrupt entfällt dieser Modus, da dieser Interrupt stets erste Priorität hat. Er führt zu einem Restart zum Speicherplatz 0066H (kann nicht vom Programmierer beeinflußt werden).

Der Modus 0 erlaubt dem unterbrechenden Gerät, jeden Befehl auf den Datenbus anzulegen zur Ausführung durch die ZVE. Beim Modus 1 führt die ZVE bei einem Interrupt einen Restart zum Speicherplatz 0038H durch.

Im Modus 2 kann durch indirekten Rufbefehl jeder beliebige Speicherplatz erreicht werden. Durch Anlegen einer 16-Bit-Adreßtabelle lassen sich Interruptroutinen an jeder beliebigen Stelle im Speicher lagern und verwalten. Dabei werden die oberen 8 Bits der jeweiligen Adresse durch den Inhalt des I-Registers (vom Programmierer voreinstellen) und die unteren vom unterbrechenden Gerät geliefert. Da bei letzterem $b_0 = 0$ sein muß, beginnen die Adressen auf geraden Speicherplätzen.

2.7.1.4.
Flags

Die beiden Flag-Register der ZVE U 880D enthalten 6 Bits mit definierter Bedeutung. Diese ist in der Befehlsliste des Anhangs angegeben.

Die Flagregister besitzen folgendes Format:

S ZxHxP/V N C

2.7.2.
Basis – MOS

Zur praktischen Programmierung steht dem Anwender des Mikrorechnersystems K 1520 eine umfangreiche Software zur Verfügung. Auf der Basis des hier beschriebenen ZVE-Befehlssatzes gibt es einen Makroassembler, der in einer gesonderten Dokumentation beschrieben wird. Weitere MOS-Komponenten werden in den folgenden Abschnitten kurz erläutert.

2.7.2.1.
K 1520 – Standardmoduln

Diese Bausteine realisieren Standardfunktionen, die nicht Bestandteil des K 1520-Befehlssatzes sind, aber zur Lösung anwenderspezifischer Probleme benötigt werden. Sie sind gleichzeitig Bestandteil weiterer MOS-Komponenten.

Ein- und Ausgabemoduln

Diese Bausteine arbeiten mit EA-Tabellen und -Rufen. Diese definieren eine Softwareschnittstelle, wodurch die Bausteine in verschiedenen Betriebssystemen verwendet werden können. Folgende Geräte können über die ADAK 6020/21 angeschlossen und bedient werden:

LBL daro 1210,

LBS daro 1215,

SD daro 1156,

BD 4000.

Über die AFS K 5121 kann ein Folienspeicher angeschlossen werden. Die ASV K 8021 ermöglicht den V 24-Anschluß.

Die Verwendung einer P ROM-Programmiereinrichtung erfolgt über PPE K 0420.

Arithmetikmoduln

Diese Bausteine verwenden folgende Datendarstellung:

- Gleitkommaformat (dual)
- Festkommaformat (dual)
  Ganzzahlige Darstellung im BCD-Code (Typ Integer).

Sie realisieren die 4 Grundrechenarten, Vergleichs- und Transportoperationen, vorzeichen- und formatspezifische Operationen. Die Genauigkeit bei FK- und GK-Zahlen ist generierbar. BCD-Zahlen werden mit einfacher (4 Byte) oder doppelter (8 Byte) Wortlänge verarbeitet.

Konvertierungsmoduln

Alle Darstellungen (intern und extern) können in beide Richtungen konvertiert werden. Die verschiedenen internen Formate sind innerhalb der Wertebereiche ineinander umrechenbar.

Mathematische Standardfunktionen

Sowohl für FK- als auch für GK-Darstellung werden Standardformate festgelegt. Dafür werden Unterprogramme für die Funktionen

$\sqrt{x}$, sin x, cos x, tan x, arc tan x, e^x, ln x, lg x, x^y

zur Verfügung gestellt.

Alle genannten Bausteine verwenden den 7-Bit-Code, den internen Binär- und den internen BCD-Code. Die Programmierung erfolgte in der Assemblersprache. Die Bausteine lassen sich beim Anwender vom Quellprogramm durch bedingte Assemblierung generieren.

Die verwendeten E/A-Moduln arbeiten im Interruptbetrieb, wodurch eine quasiparallele Bedienung peripherer Geräte möglich ist.

2.7.2.2. BASIC-Compiler K 1520

Dieser Compiler erlaubt die Ausarbeitung von Anwenderprogrammen bei einfacher Darstellung mathematisch-logischer Funktionen, geringem, aber leistungsfähigem Befehlsumfang und formalfreiem Aufbau.

Dabei wird eine BASIC-Version (ACS-BASIC) verwendet, die besonders für ökonomische Probleme vorgesehen ist (FK-Arithmetik, Magnetkonten, Stornofunktionen).

Die folgende kurze Sprachbeschreibung ersetzt nicht das Studium der entsprechenden Dokumentation.

Interne Vereinbarungen

- INTEGER-Vereinbarungen

  Ganzzahlige Variable im gepackten BCD-Format

  - 7 Stellen + Vorzeichen (4 Byte) einfaches Wort
  - 15 Stellen + Vorzeichen (8 Byte) doppeltes Wort
- ARRAY-Vereinbarung
  Feld mit wahlfreier Anzahl von INTEGER-Zahlen
- STRING-Vereinbarung
  Zeichenkette mit beliebigen druckbaren Zahlen bzw. Zeichen
- CHARACTER-Vereinbarung
  Zeichenkette der Länge 1
- EQUAL-Vereinbarung
  Gleicher Speicherbereich wird verschiedenen Namen zugewiesen
- PROCEDURE-Vereinbarung
  Vereinbarung von Unterprogrammen mit lokalen und globalen Parametern

Daten–Umwandlungen

Die Datenein- und -ausgabe erfolgt nur als Typ STRING oder CHARACTER. Für die arithmetischen Operationen sind nur Daten vom Typ INTEGER bzw. ARRAY zulässig. Deshalb sind Umwandlungen von einem in den anderen Datentyp notwendig.

- INTEGER-Konversion
  Umwandlung von STRING in INTEGER mit Überprüfung der einzelnen Zeichen durch eine Maske, für einfache und doppelte Wortlänge
- STRING-Konversion
  Umwandlung von INTEGER-Variablen einfacher oder doppelter Wortlänge in STRING mit Veränderungsmöglichkeiten für einzelne Bytes durch eine Maske
- EDIT-Konversion
  Aufbereitung von Zeichenketten: Suchen von Teilketten und Ersatz durch Austauschkette

Anweisungen zur Datenaufbereitung und -verarbeitung

- Arithmetische Anweisungen
  Addition, Subtraktion, Multiplikation, Division in Festkomma-Arithmetik nach den üblichen Vorrangregeln
- Vergleiche
  Es sind die Vergleiche $=, \neq, <, >, \leq, \geq$
  mit den logischen Verknüpfungen **AND** und **OR** zulässig
- Sprunganweisung
  GOTO<LABEL> LABEL = 1–4 Ziffern als Ansprungadresse
- Bedingte Anweisung
  IF<Bedingung>**THEN**<Anweisung>**ELSE**<Anweisung>**END**;
- FOR-Anweisung
  Mit den Schleifen **FOR...STEP...UNTIL...DO...END;** lassen sich die aus den höheren Programmiersprachen bekannten Programmschleifen konstruieren.
- Bedingte Schleifen
  WHILE...DO...END, und **REPEAT...UNTIL...END**,
  Die zwischen **DO...END**, bzw. **REPEAT...UNTIL** liegenden Anweisungen werden ausgeführt, solange die angegebene Bedingung erfüllt ist.

Dateikonzept

Die Dateien können wahlweise nur zum Lesen, nur zum Schreiben oder als Lese/Schreib-Datei konzipiert werden. Die Speicherung erfolgt in Form der sequentiellen Datei, der Direkt-Zugriffsdatei (Relativadresse) oder (als Option) KEY-Datei.

2.7.3. Betriebssysteme

2.7.3.1. Betriebssystem für MR-Entwicklungssystem MEOS 20

Dieses Betriebssystem beinhaltet die erforderlichen Systemunterlagen für das Mikrorechnerentwicklungssystem MRES 20. Es erlaubt dessen Einsatz als Prüfrechner und als Programmierarbeitsplatz.

Folgende periphere Geräte werden bedient:
Alphanumerische Tastatur
ANA 531 (BAB1)
Seriendrucker 1156
LBL 1210
LBS 1215
Folienspeicher MF 3200

Das Betriebssystem realisiert umfangreiche Funktionen:

- Anzeige und Veränderungen des ZVE-Inhalts
- Anzeige und Veränderung des RAM-Inhalts, Anzeige des P-ROM- und ROM-Inhalts
- Verschieben eines Speicherbereiches (im RAM)
- Ein- und Ausgabekorrektur
- Peripheriezuweisung
- Programmstart
- Testpunkt-Eingabe unter Berücksichtigung der beiden Testverfahren Echtzeitbetrieb und Programmtest mit TEMO-Kontrolle
- Testmonitor
- Routinen für Laden, Ein- und Ausgabe, Duplizieren, Dateiverwaltung, Kopieren von Dateien.

Das mit Hilfe des zur Verfügung stehenden Editors erstellte Programm wird durch den Assembler übersetzt in ein relativ adressiertes Objektprogramm und dabei die interne symbolische Adressierung überprüft.

Die Auflösung externer symbolischer Adreßverbindungen zwischen mehreren Programmen erfolgt durch den Binder.

2.7.3.2. Basisechtzeitsteuersystem BESI/BESE für K 1520

Dieses System ist gedacht zum Einsatz des K 1520 als Labormikrorechner und Gerätesteuerrechner. Die 2 Varianten unterscheiden sich durch Internspeicher- (BESI) und Externspeicherorientierung (BESE). Beide Steuerprogramme ermöglichen die Arbeit mit den Systemprogrammen Editor, Assembler, Binder, Lader, Testmonitor und den Dienstprogrammen.

Es werden folgende Grundfunktionen realisiert:

- Vorrangorganisation (Multi-Task-Funktion),
- Interruptorganisation,
- Zeitorganisation,
- Unterprogrammorganisation,
- Ein- und Ausgabeorganisation und Bedienerkommunikation,
- Dateisteuerung und -verwaltung,
- Externe Programmsteuerung und -verwaltung,
- Funktionskomponenten für den Systemstart und Ein- und Ausgabekonvertierung.

2.7.4. Wirtsrechner – MOS (Cross-Software)

Zur Aufbereitung, Übersetzung und Testung von Programmen für den K 1520 steht für verschiedene Wirtsrechner entsprechende Software zur Verfügung.

Für die Rechenanlagen des ESER, den R 4000 und den KRS 4200, gibt es jeweils einen

- Cross-Assembler (CRAS z. B. 4000-K 1520)
- Cross-Binder (CRBI)
- Cross-Simulations- und Testsystem.

Für den R 4000 steht ferner ein plattenorientiertes Cross-Aufbereitungssystem und für den KRS 4200 ein solches auf Magnetbandbasis zur Verfügung (CRAP).

3. Sonstiges

3.1. Umgebungsbedingungen

Die Werte für die Umgebungsbedingungen dürfen die Einsatzklasse EKL 5/60/30/95/10-1_E für die Baugruppen ZRE, OPS, OFS, PFS, BVE, ABD, ADA nicht überschreiten. Oberhalb +40 °C ist eine Zwangsbelüftung erforderlich.

Die Bedieneinheit wurde für Umgebungsbedingungen für die Einsatzklasse EKL 5/40/30/95/10-1_E konstruiert.

3.2. Netzanschluß der Stromversorgungsmoduln K 1510

Werden zur Stromversorgung des Mikrorechners K 1520 die Stromversorgungsmoduln des Mikrorechnersystems K 1510 eingesetzt, ist ihr Netzanschluß an 220 V + 10%, —15%, Einphasenwechselspannung 50/60 Hz ±1Hz, zu realisieren. Netzausfälle bis 10 ms werden von den Stromversorgungsmoduln überbrückt. Bei der Dimensionierung der Stromversorgungsmodule für eine bestimmte Rechnerkonfiguration K 1520 ist zu beachten, daß die Stromversorgungsmoduln prinzipiell nicht parallel geschaltet werden dürfen.

3.3. Dimensionierung der Stromversorgung

In der Beschreibung der einzelnen Funktionsgruppen sind die Angaben für die benötigten Spannungen und dazugehörigen Leistungsaufnahmen enthalten. Bei der Dimensionierung der Stromversorgung aus den Modulen des Mikrorechnersystems K 1510 sind die entsprechenden Dimensionierungs- bzw. Projektierungsrichtlinien zu beachten.

3.4. Programmierung der EPROM

Zur Programmierung der EPROM U 555 ist die PROM-Programmiereinrichtung PPE K 0420 zu verwenden. Diese wird im Teil II der Systembeschreibung vorgestellt. Im Mikrorechnerentwicklungssystem MRES 20 ist eine solche PPE integriert, so daß mit diesem System ebenfalls die Programmierung der EPROM U 555 erfolgen kann.

3.5. Lieferformen

Der K 1520 wird als in Finalgeräte einbaubarer „OEM"-Mikrorechner ausgeliefert. Folgende Lieferformen sind vorgesehen:

- Steckeinheitenmoduln
- Steckeinheiteneinsätze mit Steckeinheitenmoduln in den Verkaufsformen:
  - OEM-Single-Board-Rechner, eine Steckeinheit ZRE (endgeprüft)
  - OEM-Baugruppenverbund, ein Satz Steckeinheiten (endgeprüft)
  - OEM-Baugruppenrechner, als mit Steckeinheiten bestückter Steckeinheiteneinsatz (endgeprüft)
- Systemunterlagen K 1520:
  - Cross-MOS zur Programmaufbereitung auf Rechnern der Familie 4000 und des ESER
  - Betriebssysteme
  - Basis-MOS (Standardmoduln)

Dokumentation gehört nicht zum Lieferumfang! Die Betriebsdokumentation K 1520 muß ebenfalls gesondert bestellt und bezogen werden.

3.6. Absatz

Der Verkauf des Mikrorechnersystems K 1520 gemäß 3.5. und der Baugruppenübersicht in der Anlage erfolgt für die DDR durch

VEB Robotron-Vertrieb Berlin
Abteilung VA 5
108 Berlin
Mohrenstraße 62

Die Moduln des Mikrorechners K 1510 können ebenfalls über diesen Absatzbetrieb bezogen werden.

Für den Export ist der Außenhandelsbetrieb

Robotron Export-Import
1086 Berlin
PSF 1235

zuständig. Über alle Leistungen zum Mikrorechnersystem K 1520 gibt Ihnen der jeweilige Absatzbetrieb umfassende Auskünfte. Die Bestellung und der Bezug peripherer Geräte, die an den Mikrorechner K 1520 angeschlossen werden können, erfolgen für die Anwender aus der DDR bei dem territorial zuständigen Robotron-Vertriebsbetrieb.

Für die Anwender außerhalb der DDR erfolgen die Bestellung und der Bezug ebenfalls über Robotron Export-Import. Über die vorgesehenen Kundendienstleistungen können Sie sich beim Kauf bzw. vorher in einem unverbindlichen Informationsgespräch beim Absatzbetrieb informieren.

3.7. Systemkomplettierung

In der Systembeschreibung Mikrorechnersystem robotron K 1520, Teil I, sind die Grundbaugruppen beschrieben. Im Teil II werden Ergänzungsbaugruppen vorgestellt.

Im einzelnen sind zur Systemkomplettierung folgende Bau- und Funktionsgruppen vorgesehen:

- Zentrale Recheneinheit ZRE K 2525 mit Zähler/Zeitgeber, Taktgenerator und 8 K Byte EPROM, aber ohne Mehrrechnerkopplungsinterface
- Operativspeicher OPS K 3521 mit 4 K Byte statisch RAM (c-MOS) und Spannungsunterstützung bei Netzausfall durch Batterie, es werden etwa 200 h Datenerhalt gewährleistet
- Operativspeicher OPS K 3525 mit 16 K Byte dynamisch RAM, die durch den systeminternen Refresh wie statische RAM behandelt werden
- Operativ-/Festwertspeicher OFS K 3621 mit 2 K Byte statisch RAM (c-MOS) und Datenerhalt wie bei OPS K 3521 sowie 6 K Byte EPROM
- Anschlußsteuerung für Folienspeicher (1···4 Laufwerke) MF 3200 mit der Kurzbezeichnung AFS K 5121
- Anschlußsteuerung für Bildschirmanzeigebaugruppe (BAB 1) ABS K 7023
- Anschlußsteuerung für Tastatur und Drucker ATD K 7026
- Anschlußsteuerung für V 24-Interface (3···4 Kanäle) ASV K 8021
- Anschlußsteuerung für daro-Geräte mit Standardinterface sif 1000 mit 1 Eingabe- und 1 Ausgabekanal, Statusregister und programmierbarer RUF-Abschaltung ADA K 6022.
  Diese Anschlußsteuereinheit löst die Anschlußsteuereinheiten ADA K 6020 und K 6021 ab.
- EPROM-Programmiereinrichtung PPE K 0420 für U 555 oder äquivalente PROM-Speicherschaltkreise

- EPROM-Löschgerät PLG K 0421 als autonomes Auftischgerät zur Löschung der auf den o. g. EPROM eingeschriebenen Informationen
- EPROM-Aufnahmeeinrichtung PRA K 0422 zur Aufnahme o. g. EPROM

Zur Unterstützung der Anwender des Mikrorechnersystems K 1520 und der Anwender der ZVE U 880 D bei der Programmentwicklung, Programmtestung, EPROM-Programmierung und im Einsatz als Prüfrechner dient das Mikrorechnerentwicklungssystem robotron MRES 20. Weitere Einsatzmöglichkeiten dieses auf Basis des K 1520 aufgebauten Entwicklungssystems sind möglich. Dieses autonome System in einer praktischen Gefäßlösung wird in der Systembeschreibung Teil III vorgestellt.

Die Entwicklung weiterer Systemkomponenten zur Vervollständigung des Mikrorechnersystems K 1520 ist vorgesehen.

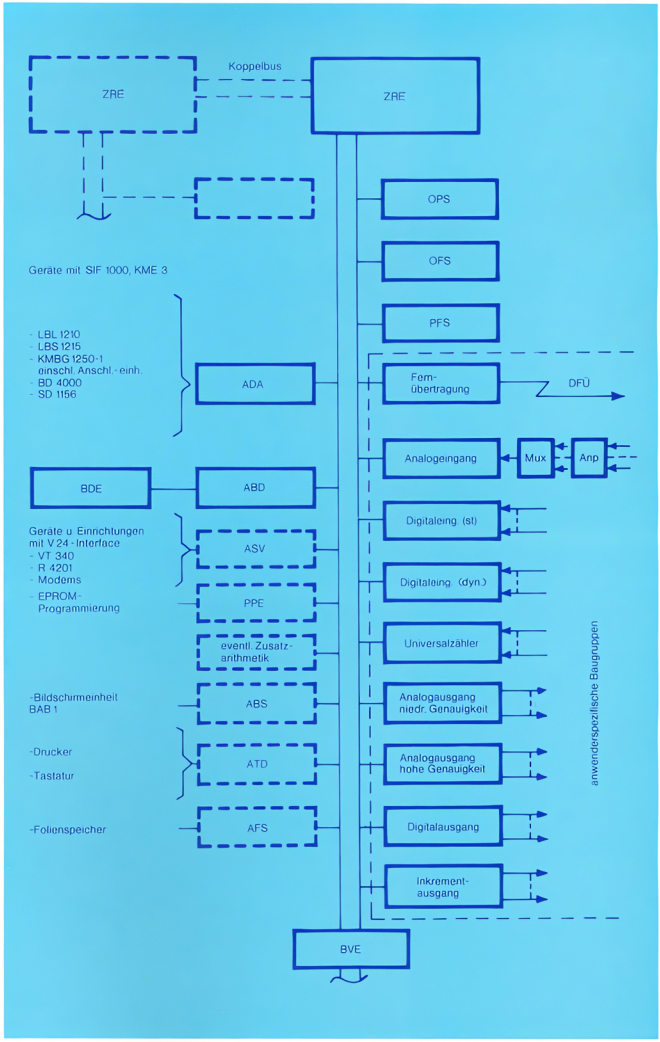
Änderungen und Ergänzungen im Sinne des technischen Fortschrittes bleiben vorbehalten und werden rechtzeitig bekanntgegeben.

Übersicht der Gerätekomponenten Mikrorechner K 1520

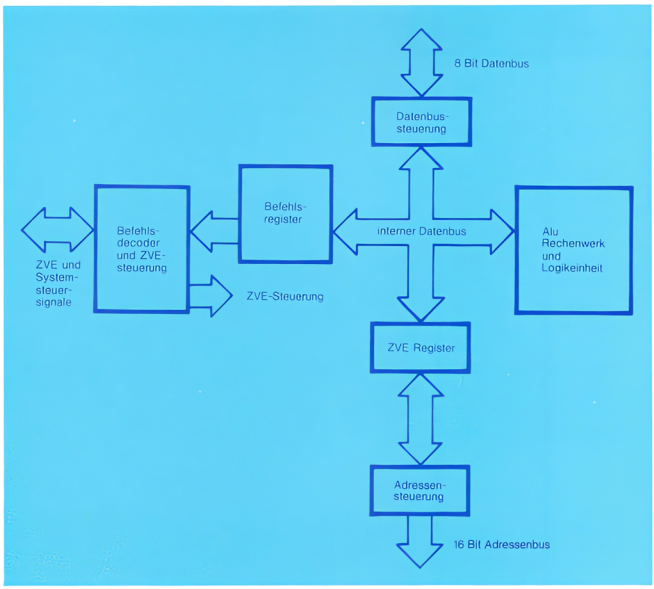
Bezeichnung	Kurzbez.	Chiffre
Zentrale Recheneinheit kpl.	ZRE	K 2521
Zentrale Recheneinheit ohne Taktgenerator	ZRE	K 2522
Zentrale Recheneinheit ohne Zähler/Zeitgeber	ZRE	K 2523
Zentrale Recheneinheit ohne Taktgenerator und ohne Zähler/Zeitgeber	ZRE	K 2524
Operativspeicher, 4 K RAM (n-MOS)	OPS	K 3520
Operativ-/Festwertspeicher 2 K RAM, 6 K EPROM	OFS	K 3620
Festwertspeicher 16 K EPROM	PFS	K 3820
Busverstärker	BVE	K 4120
Anschlußsteuerung für daro-Geräte	ADA	K 6020
Anschlußsteuerung für daro-Geräte	ADA	K 6021
Anschlußsteuerung Bedieneinheit	ABD	K 7022
Bedieneinheit	BDE	K 7622
Verbindungsleitung	LTG	K 0521
Verbindungsleitungsadapter	VLA	K 0522
Steckeinheiteneinsatz 5 Steckplätze	STS	K 0120
Steckeinheiteneinsatz 11 Steckplätze	STS	K 0121
In der weiteren Komplettierung sind vorgesehen:		
Operativspeicher 4 K RAM (c-MOS)	OPS	K 3521
Operativspeicher 16 K dyn. RAM	OPS	K 3525
Operativ-/Festwertspeicher 2 K RAM (c-MOS) 6 K EPROM	OFS	K 3621
Zentrale Recheneinheit 8 K EPROM, ohne Mehrrechnerkoppelinterface	ZRE	K 2525
Anschlußsteuerung Folienspeicher	AFS	K 5121
Anschlußsteuerung V 24	ASV	K 8021
Anschlußsteuerung Bildschirm BAB 1	ABS	K 7023
Anschlußsteuerung Tastatur und Drucker	ATD	K 7026
EPROM-Programmiereinrichtung	PPE	K 0420
EPROM-Löschgerät	PLG	K 0421
EPROM-Aufnahmeeinrichtung	PRA	K 0422

Mikrorechner K 1520 – Gerätekonfiguration

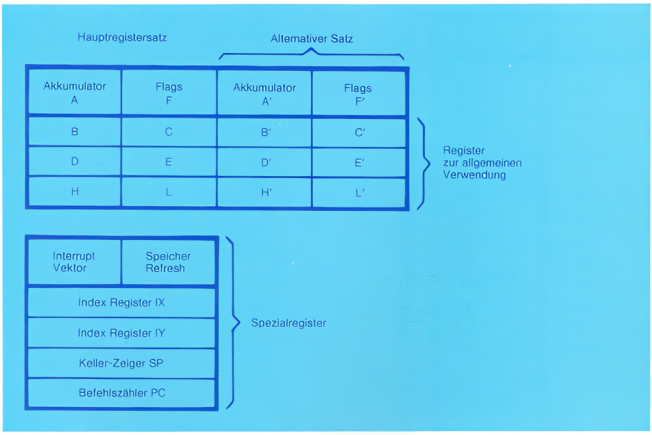
(Beispiel mit Einbeziehung von anwenderspezifischen Baugruppen)



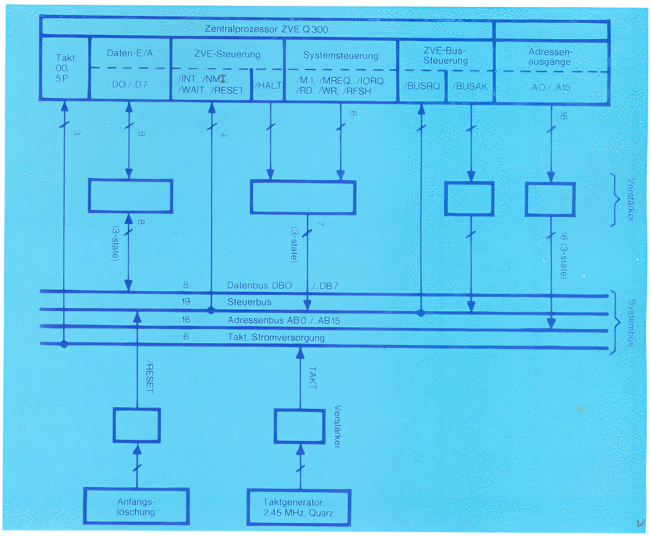
Blockschaltbild ZVE U 880 D



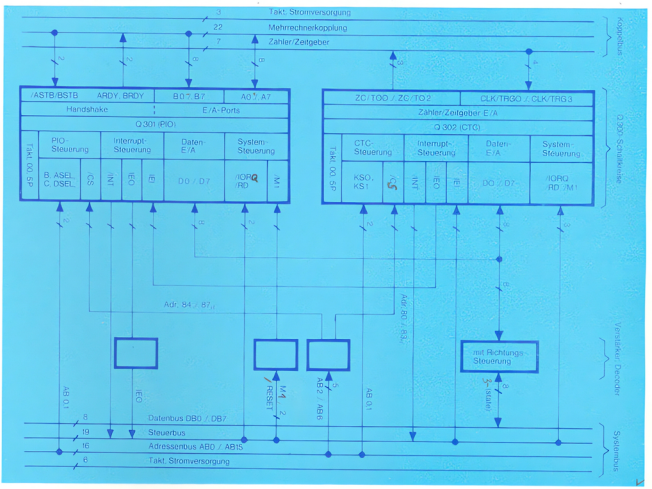
Registeraufbau ZVE U 880 D



Übersicht
Systembusgenerierung
mit Zentralprozessor,
Anfangslöschung und
Taktgenerator
(auf K 2521 bis K 2524)

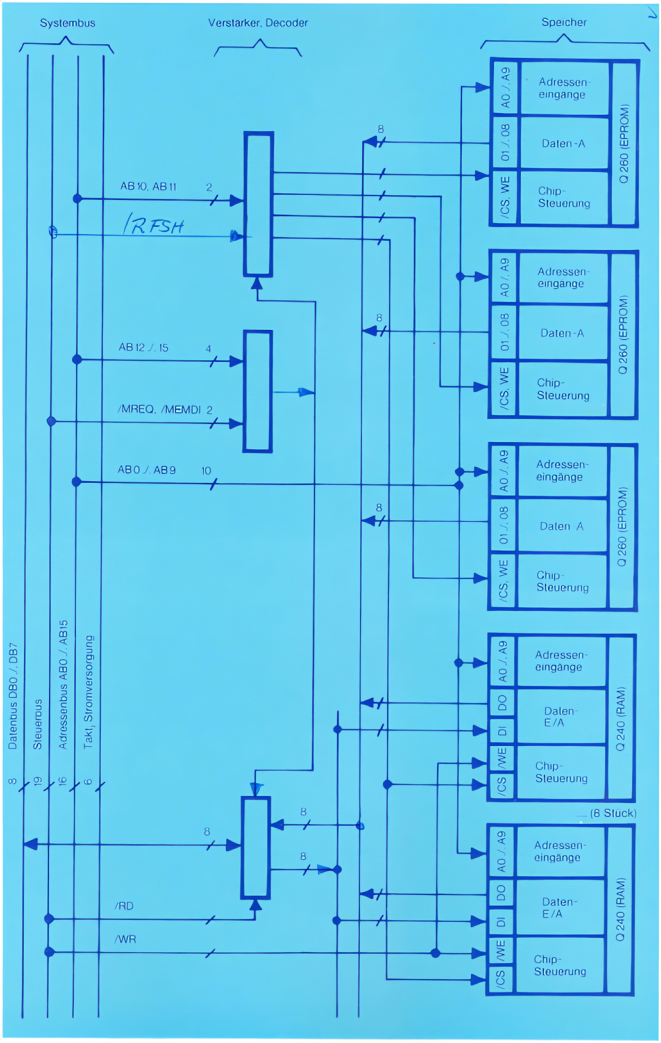


Anschluß peripherer
Schaltkreise an Systembus
und Koppelbus (auf K 2521)



Speicheranschluß an Systembus

Beispiel: 1 K Byte RAM, 3 K Byte EPROM (auf K 2521 bis K 2524)



Systembus

PIN	B	A
29	5 P	5 P
28	12 P	12 P
27	/BAI	/BA0
26	/HALT	/M1
25	/RDY	/RFSH
24	/IORQ	/WAIT
23	/INT	/NMI
22	00	/IODI
21	00	TAKT
20	/BUSRQ	/RESET
19	AB 1	AB 0
18	AB 3	AB 2
17	AB 5	AB 4
16	AB 7	AB 6
15	(5N)	5N
14	AB 9	AB8
13	AB 11	AB10
12	AB 13	AB12
11	AB 15	AB14
10	/IEI	/IEO
9	/MEMDI	/MREQ
8	/RD	/WR
7	DB0	DB1
6	DB2	DB3
5	DB4	DB5
4	DB6	DB7
3	5PG	5PG
2	00	00
1	00	00

Koppelbus

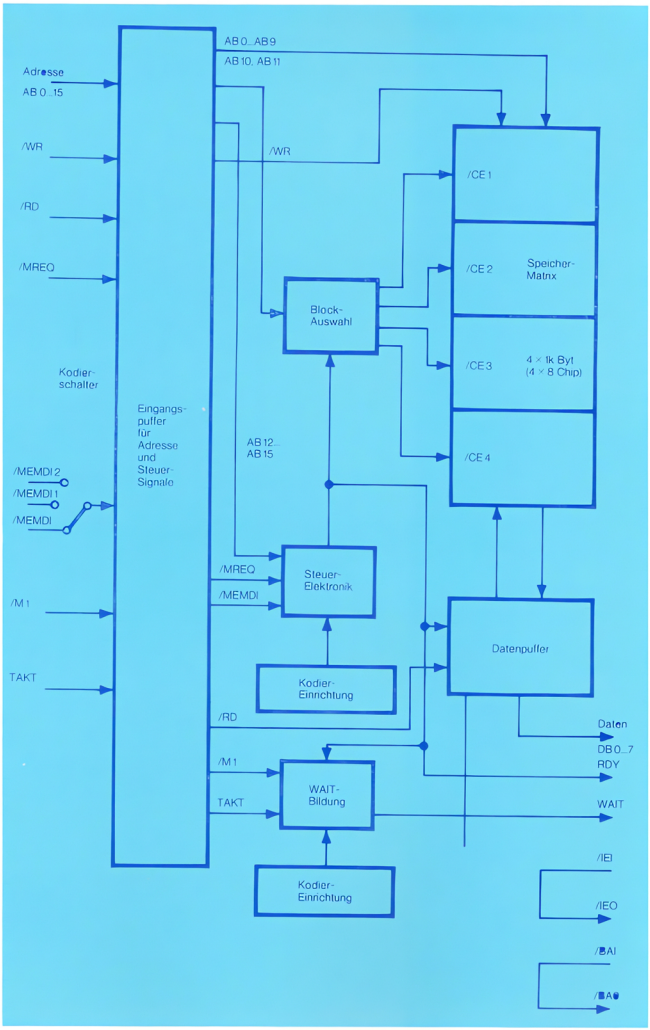
PIN	B	A
29	00	00
28	00	00
27	—	—
26	—	—
25	CLK/TRG0	ZC/T00
24	CLK/TRG1	ZC/T01
23	CLK/TRG2	ZC/T02
22	SUE	CLK/TRG3
21	/MEMDI2	/MEMDI1
20	—	/MEMAS
19	/MEMPR	/IOAS
18	/RDYP	/IOPR
17	/PASTB	/PBSTB
16	PA1	PA0
15	PA3	PA2
14	PA5	PA4
13	PA7	PA6
12	PARDY	PBRDY
11	PB1	PB0
10	PB3	PB2
9	PB5	PB4
8	PB7	PB6
7	/KOPL	CREIT
6	/PRDY	/PSTB
5	/RESET	00
4	TAKT	00
3	U_s	U_s
2	U_s	U_s
1	5P	5P

Systembus: B 15 darf auf der Steckeinheit nicht mit A 15 gebrückt sein!

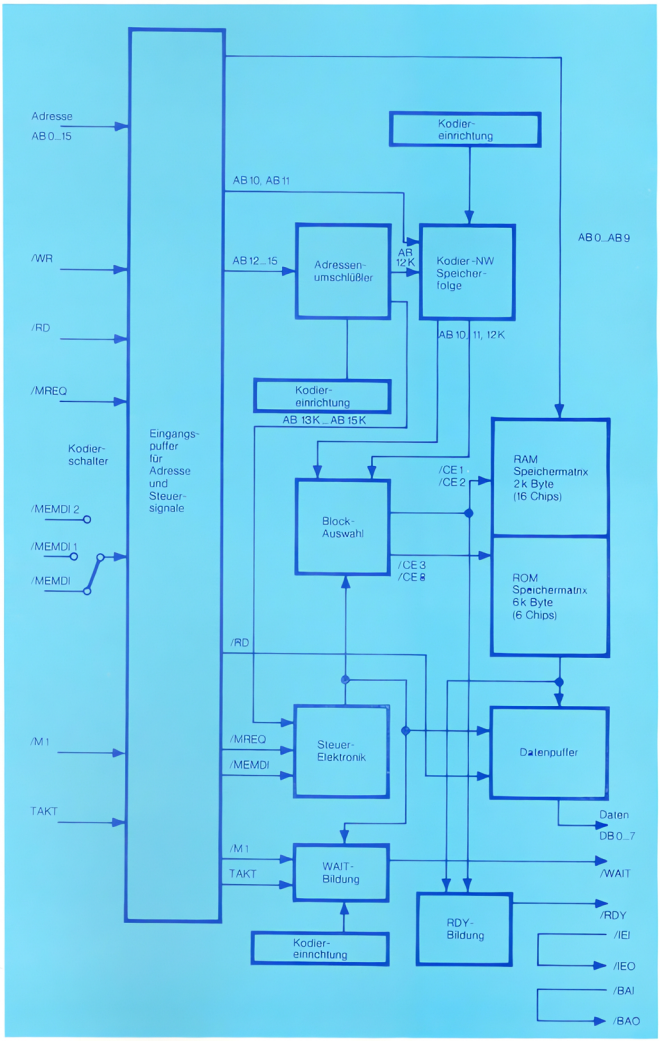
Koppelbus:

A/B 4–6, A/B 8–17	Mehrrechnerkopplung (bei ZRE belegt)
A/B 2–3, A/B 26–27	anwenderspezifisch belegt
A 22, A/B 23–25	Zähler/Zeitgeber-Signal (bei ZRE belegt)
A/B 7, A/B 18–19, A 20	von Busverstärker belegt (bei Prüfrechner)
B 22	Spannungsüberwachung (bei c-MOS-Speichern belegt)
A/B 21	Speichererweiterung über 64 K (bei Speichersteckeinheiten belegt)

Blockschaltbild OPS K 3520



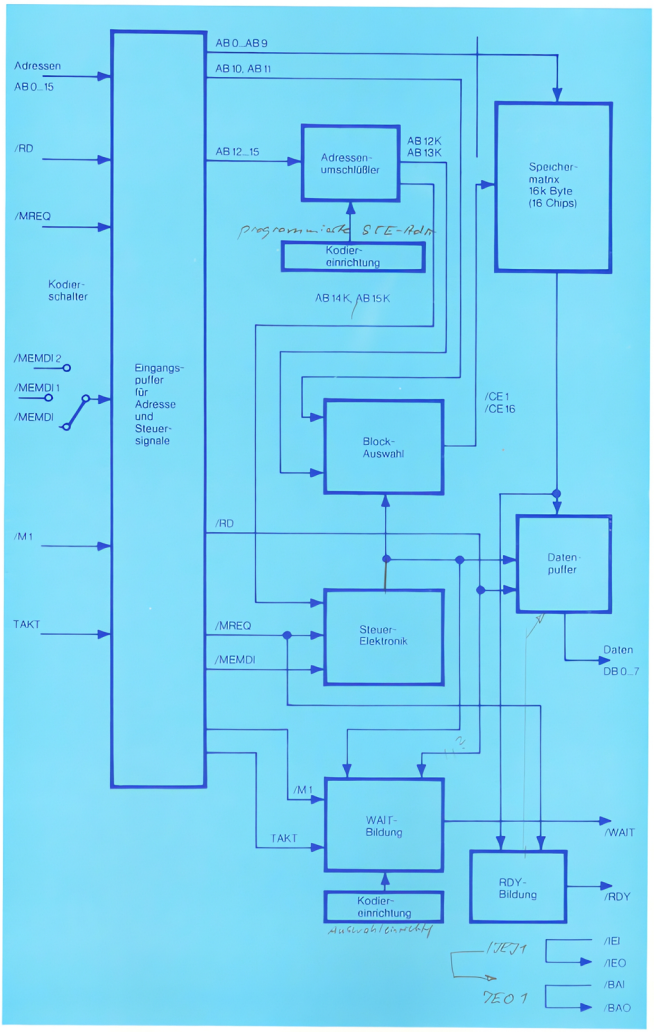
Blockschaltbild OFS K 3620

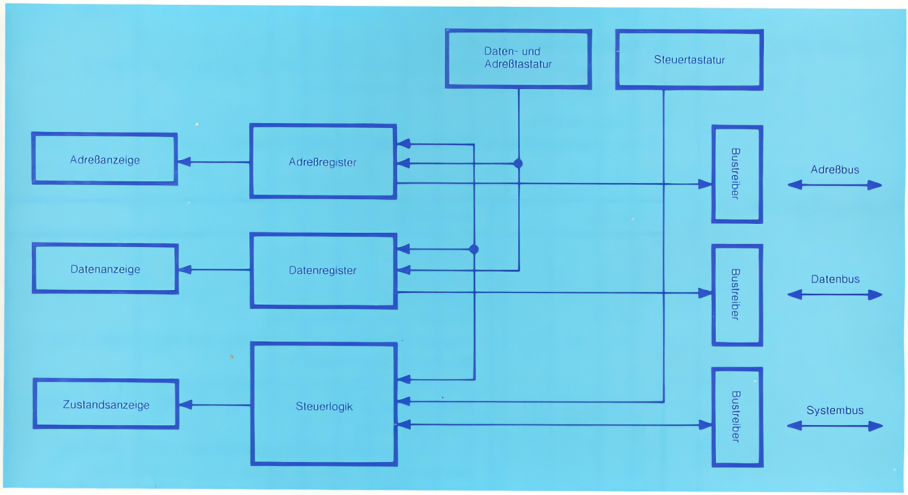


Blockschaltbild PFS K 3820 (BDZ)

012 - 7041

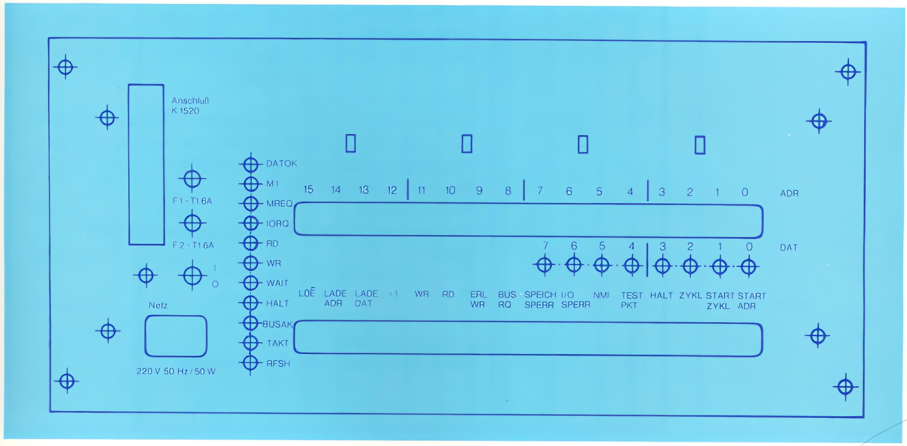
vgl. Betriebsdok. Mitteleckessystem K1520 IV-24





Blockschaltbild Bedieneinheit

Bedien- und Anzeigefeld der Bedieneinheit



Befehlszeilenaufbau:

- Symbolische Adressen
  Symbolische Adressen können maximal 5 Zeichen lang sein. Zugelassen sind dabei Buchstaben und Ziffern, die linksbündig einzugeben sind. Das erste Zeichen muß ein Buchstabe sein. Abgeschlossen wird die symbolische Adresse durch einen Doppelpunkt.
- Operationsfeld
  Das Operationsfeld enthält die Befehlsmnemonik oder den Pseudobefehl. Es wird durch Tabulator oder mindestens ein Leerzeichen vom Operandenfeld getrennt.
- Operandenfeld
  Das Operandenfeld enthält den oder (durch Komma getrennt) die Operanden.
- Kommentarfeld
  Das Kommentarfeld beginnt mit einem Semikolon. Zugelassen sind maximal 61 Zeichen zusammen mit dem Operandenfeld.
- Zeilenabschluß
  Die Befehlszeile wird mit CR, LF oder NL abgeschlossen.

Mnemonik ZVE U 880 D:

Symbole	Bedeutung
C	Carry-Flag
Z	Zero-Flag
S	Signum-Flag
P/V	Parität- oder Überlauf-Flag Logische Operationen beeinflussen das Flag durch Parität des Ergebnisses P. Arithmetische Operationen beeinflussen das Flag durch Überlauf des Ergebnisses V.
H	Half-Carry-Flag H = 1, wenn die Addition (Subtraktion) einen Übertrag (Borgen) in das (vom) 5. Bit des Akkumulators liefert.
N	Additions-/Subtraktions-Flag N = 1, wenn die vorhergehende Operation eine Subtraktion war.
M	Der durch (HL) adressierte Speicherplatz; dafür kann auch (HL) geschrieben werden.
PC	Programmzähler
IX, IY	Indexregister (16 Bit) zur Adressierung von Datenbereichen.
SP	Stackpointer H-Teil 1. Register L-Teil 2. Register
AF	Akkumulator und Flags
(IX+d), (IY+d)	Speicherplatz, der durch Indexregister und Verschiebung adressiert wird. Bei der Berechnung der Adresse wird der Wert des Indexregisters nicht verändert.
.	Flag wird nicht beeinflußt
X	Flag unbestimmt Flag wird entsprechend dem Ergebnis der Operation gesetzt
①	P/V = 0, wenn BC = 0, sonst P/V = 1
②	Z = 1, wenn A = M, sonst Z = 0
③	Z = 1, wenn B = 0, sonst Z = 0
←	wird geladen nach
( )	Inhalt des Registers oder Speicherplatzes

Mnemonik	Symbolische Operationen	Flags						Op.-Code			Byte	Masch.-Zyklen	Takte	Bemerkung
		C	Z	P/V	S	N	H	76	543	210				
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
LD r_1, r_2	$r_1 \leftarrow r_2$	.	.	.	.	.	.	01	r_1	r_2	1	1	4	
LD r, n	$r \leftarrow n$	.	.	.	.	.	.	00 ←	r	110 →	2	2	7	
								←	n	→				
LD r, M	$r \leftarrow M$	.	.	.	.	.	.	01	r	110	1	2	7	
LD r, (IX+d)	$r \leftarrow (IX+d)$	.	.	.	.	.	.	11	011	101	3	5	19	
								01	r	110				
								←	d	→				r, r_1, r_2 ZVE-Register
LD r, (IY+d)	$r \leftarrow (IY+d)$	.	.	.	.	.	.	11	111	101	3	5	19	000 B
								01	r	110				001 C
								←	d	→				010 D
LD M, r	$M \leftarrow r$	.	.	.	.	.	.	01	110	r	1	2	7	011 E
LD (IX+d), r	$(IX+d) \leftarrow r$	.	.	.	.	.	.	11	011	101	3	5	19	100 H
								01	110	r				101 L
								←	d	→				111 A
LD (IY+d), r	$(IY+d) \leftarrow r$	.	.	.	.	.	.	11	111	101	3	5	19	
								01	110	r				
								←	d	→				
LD M, n	$M \leftarrow n$	.	.	.	.	.	.	00	110	110	2	3	10	
								←	n	→				
LD (IX+d), n	$(IX+d) \leftarrow n$	.	.	.	.	.	.	11	011	101	4	5	19	
								←	d	→				
								←	n	→				
LD (IY+d), n	$(IY+d) \leftarrow n$	.	.	.	.	.	.	11	111	101	4	5	19	
								00	110	110				
								←	d	→				
								←	n	→				
LD A, (BC)	$A \leftarrow (BC)$	.	.	.	.	.	.	00	001	010	1	2	7	
LD A, (DE)	$A \leftarrow (DE)$	.	.	.	.	.	.	00	011	010	1	2	7	
LD A, (nn)	$A \leftarrow (nn)$	.	.	.	.	.	.	00	111	010	3	4	13	
								←	n	→				
								←	n	→				
LD (BC), A	$(BC) \leftarrow A$	.	.	.	.	.	.	00	000	010	1	2	7	
LD (DE), A	$(DE) \leftarrow A$	.	.	.	.	.	.	00	010	010	1	2	7	
LD (nn), A	$(nn) \leftarrow A$	.	.	.	.	.	.	00	110	010	3	4	13	
								←	n	→				
								←	n	→				
LD A, I	$A \leftarrow I$	.	↕	IFF	↕	0	0	11	101	101	2	2	9	
								01	010	111				
LD A, R	$A \leftarrow R$	.		IFF		0	0	00	101	101	2	2	9	
								01	011	111				
LD I, A	$I \leftarrow A$	.	.	.	.	.	.	11	101	101	2	2	9	
								01	000	111				
LD R, A	$R \leftarrow A$	.	.	.	.	.	.	11	101	101	2	2	9	
								01	001	111				
LD dd, nn	$dd \leftarrow nn$	.	.	.	.	.	.	00	dd0	001	3	3	10	dd Reg.-Paar
								←	n	→				00 BC
								←	n	→				01 DE

Mnemonik	Symbolische Operationen	Flags						Op.-Code			Byte	Masch.-Zyklen	Takte	Bemerkung
		C	Z	P/V	S	N	H	76	543	210				
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
LD IX, nn	IX ← nn	.	.	.	.	.	.	11 ← ←	011 n n	101 → →	4	4	14	10 HL 11 SP
LD IY, nn	IY ← nn	.	.	.	.	.	.	11 00 ← ←	111 100 n n	101 001 → →	4	4	14	
LD HL, (nn)	H ← (nn+1) L ← (nn)	.	.	.	.	.	.	00 ← ←	101 n n	010 → →	3	5	16	
LD dd, (nn)	dd_H ← (nn+1) dd_L ← (nn)	.	.	.	.	.	.	11 01 ← ←	101 dd1 n n	101 011 → →	4	6	20	
LD IX, (nn)	IX_H ← (nn+1) IX_L ← (nn)	.	.	.	.	.	.	11 00 ← ←	011 101 n n	101 010 → →	4	6	20	
LD IY, (nn)	IY_H ← (nn+1) IY_L ← (nn)	.	.	.	.	.	.	11 00 ← ←	111 101 n n	101 010 → →	4	6	20	
LD (nn), HL	(nn+1) ← H (nn) ← L	.	.	.	.	.	.	00 ← ←	100 n n	010 → →	3	5	16	
LD (nn), dd	(nn+1) ← dd_H (nn) ← dd_L	.	.	.	.	.	.	11 01 ← ←	101 dd0 n n	101 011 → →	4	6	20	
LD (nn), IX	(nn+1) ← IX_H (nn) ← IX_L	.	.	.	.	.	.	← ←	n n	→ →				
LD (nn), IY	(nn+1) ← IY_H (nn) ← IY_L	.	.	.	.	.	.	11 00 ← ←	111 100 n n	101 010 → →	4	6	20	
LD SP, HL	SP ← HL	.	.	.	.	.	.	11	111	011	1	1	6	
LD SP, IX	SP ← IX	.	.	.	.	.	.	11 11	011 111	011 001	2	2	10	
LD SP, IY	SP ← IY	.	.	.	.	.	.	11 11	111 111	101 001	2	2	10	
PUSH qq	(SP-2) ← qq_L (SP-1 ← qq_H	.	.	.	.	.	.	11	qq0	101	1	3	11	qq Reg.-Paar 00 BC 01 DE 10 HL 11 AF
PUSH IX	(SP-2) ← IX_L (SP-1) ← IX_H	.	.	.	.	.	.	11 11	011 100	101 101	2	4	15	
PUSH IY	(SP-2) ← IY_L (SP-1) ← IY_H	.	.	.	.	.	.	11 11	111 100	101 101	2	4	15	
POP qq	qq_H ← (SP+1) qq_A ← (SP)	.	.	.	.	.	.	11	qq0	001	1	3	10	

Mnemonik	Symbolische Operationen	Flags						Op.-Code			Byte	Masch.-Zyklen	Takte	Bemerkung
		C	Z	P/V	S	N	H	76	543	210				
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
POP IX	$IX_H \leftarrow (SP+1)$ $IX_L \leftarrow (SP)$	.	.	.	.	.	.	11 11	011 100	101 001	2	4	14	
POP IY	$IY_H \leftarrow (SP+1)$ $IY_L \leftarrow (SP)$	.	.	.	.	.	.	11 11	111 100	101 001	2	4	14	
EX DE, HL	DE ← HL	.	.	.	.	.	.	11	101	011	1	1	4	
EXAF	AF AF	.	.	.	.	.	.	00	001	000	1	1	4	Austausch der Registersätze AF, BC, DE, HL mit den Hilfsregistersätzen AF', BC', DE', HL'
EXX	BC BC' DE DE' HL HL'	.	.	.	.	.	.	11	011	001	1	1	4	
EX (SP), HL	H (SP+1) L (SP)	.	.	.	.	.	.	11	100	011	1	5	19	
EX (SP), IX	IXH (SP+1) IXL (SP)	.	.	.	.	.	.	11 11	011 100	101 011	2	6	23	
EX (SP), IY	IYH (SP+1) IYL (SP)	.	.	.	.	.	.	11 11	111 100	101 011	2	6	23	
LDI	(DE) ← M DE ← DE+1 HL ← HL+1 BC ← BC—1	.	?	↕ ①	?	0	0	11 10	101 100	101 000	2	4	16	Lade den durch Registerpaar DE adressierten Speicherplatz mit Inhalt des durch HL adressierten Speicherplatzes, erhöhe DE und HL um 1 und erniedrige BC um 1
LDIR	(DE) ← M DE ← DE+1 HL ← HL+1 BC ← BC+1 wiederholen bis BC = 0	.	?	0	?	0	0	11 10	101 110	101 000	2 2	5 4	21 16	wenn BC ≠ 0 wenn BC = 0
LDD	(DE) ← M DE ← DE—1 HL ← HL—1 BC ← BC—1	.	?	↕ ①	?	0	0	11 10	101 101	101 000	2	4	16	
LDDR	(DE) ← M DE ← DE—1 HL ← HL—1 BC ← BC—1 wiederholen bis BC = 0	.	? ②	0 ①	?	0	0	11 10	101 111	101 000	2 2	5 4	21 16	wenn BC ≠ 0 wenn BC = 0
CPI	A′– M HL ← HL+1 BC ← BC—1	.	↕ ②	↕ ①	?	1	?	11 10	101 100	101 001	2	4	16	
CPIR	A′– M HL ← HL+1	.	↕	↕	?	1	?	11 10	101 110	101 001	2	5 4	21 16	wenn BC ≠ C und A ≠ M wenn BC = 0 oder A = M

Mnemonik	Symbolische Operationen	Flags						Op.-Code			Byte	Masch.-Zyklen	Takte	Bemerkung
		C	Z	P/V	S	N	H	76	543	210				
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	BC ← BC—1 wiederholen, bis A = M oder BC = 0		②	①										
CPD	A – M HL ← HL—1 BC ← BC—1		↕	↕	?	1	?	11 10	101 101	101 001	2	4	16	
CPDR	A = M		② ↕	① ↕	?	1	?	11	101	101	2	5	21	wenn BC ≠ 0 und A ≠ M
	HL ← HL—1							10	111	001	2	4	16	wenn BC = 0 oder A = M
	BC ← BC—1 wiederholen, bis A = M oder BC = 0													
ADD r	A ← A+r	↕	↕	V	↕	0	↕	10	000	r	1	1	4	r ZVE-Register 000 B 001 C 010 D 011 E 100 N
ADD n	A ← +n	↕	↕	V	↕	0	↕	11 ←	000 n	110 →	2	2	7	
ADD M	A ← A+M	↕	↕	V	↕	0	↕	10	000	110	1	2	7	101 L 111 A
ADD (IX+d)	A ← A+(IX+d)	↕	↕	V	↕	0	↕	11 10 ←	011 000 d	101 110 →	3	5	19	
ADD (IY+d)	A ← A+(IY+d)	↕	↕	V	↕	0	↕	11 10 ←	111 000 d	101 110 →	3	5	19	
ADC s	A ← A+s+CY	↕	↕	V	↕	0	↕		001					s ist entweder r, n, M, (IX+d), (IY+d) analog den ADD-Befehlen
SUB s	A ← A—s	↕	↕	V	↕	1	↕		010					
SBC s	A ← A—s—CY	↕	↕	V	↕	1	↕		011					die umrandeten Bits ersetzen OOO im ADD-Satz
AND s	A ← A s	0	↕	P	↕	0	1		100					
OR s	A ← A s	0	↕	P	↕	0	0		110					
XOR s	A ← A s	0	↕	P	↕	0	0		101					
CMP s	A – s	↕	↕	V	↕	1	↕		111					
INC r	r ← r+1	·	↕	V	↕	0	↕	00	r	100	1	1	4	
INC M	M ← M+1	·	↕	V	↕	0	↕	00	110	100	1	3	11	
INC (IX+d)	(IX+d) ← (IX+d)+1	·	↕	V	↕	0	↕	11 00 ←	011 110 d	101 100 →	3	6		
INC (IY+d)	(IY+d) ← (IY+d)+1	·	↕	V	↕	0	↕	11 00 ←	111 110 d	101 100 →	3	6		

Mnemonik	Symbolische Operationen	Flags						Op.-Code			Byte	Maschinen-Zyklen	Takte	Bemerkung
		C	Z	P/V	S	N	H	76	543	210				
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
DEC f	f ← f — 1	.	↕	V	↕	1	↕		101					f ist entweder r, M, (IX+d), (IY+d) analog den INC-Befehlen Format und Zustände wie bei INC-Befehlen Austausch von 100 mit 101 im Op.-Code
DAA	Umwandlung des Akku-Inhaltes in gepackte BCD-Zahl, anschl. an Add. oder Sub. mit gepackten BCD-Operanden	↕	↕	P	↕		↕	00	100	111	1	1	4	
CPL	A ← /A	.	.	.	.	1	1	00	101	111	1	1	4	Bitweise Komplementierung des Akkumulatorinhaltes
NEG	A ← /A+1	↕	↕	V	↕	1	↕	11 01	101 000	101 100	2	2	8	Komplementierung des Akkumulatorinhaltes und +1 (Zweierkomplement)
CCF	CY ← /CY	↕	.	.	.	0	×	00	111	111	1	1	4	Komplementierung des Carry-Flags
SCF	CY ← 1	1	.	.	.	0	0	00	110	111	1	1	4	Setze Carry-Flag auf 1
NOP	keine Operat.	.	.	.	.	.	.	00	000	000	1	1	4	
HALT	ZVE im Stopzustand	.	.	.	.	.	.	01	110	110	1	1	4	
DI	IFF ← 0	.	.	.	.	.	.	11	110	011	1	1	4	
EI	IFF ← 1	.	.	.	.	.	.	11	111	011	1	1	4	
IM∅		.	.	.	.	.	.	11 01	101 000	101 110	2	2	8	Ausführung eines Befehls, der über Datenbus geliefert wird. Befehl kommt vom unterbrechenden Gerät
IM1		.	.	.	.	.	.	11 01	101 010	101 110	2	2	8	Bei Interrupt-Ausführung von RST 38 H
IM2		.	.	.	.	.	.	11 01	101 011	101 110	2	2	8	Interrupt durch indirekten Aufruf auf beliebigen Speicherplatz. Bei Interrupt wird ein indirekter Call ausgeführt, wobei der höherwertige Teil der Adresse aus I, der niederwertige Teil vom Datenbus kommt
ADD HL, dd	HL ← HL+dd	↕	.	.	.	0	×	00	dd1	001	1	3	11	dd Reg.-Paar 00 BC 01 DE
ADC HL, dd	HL ← HL+dd +CY	↕	↕	V	↕	0	×	11 01	101 dd1	101 010	2	5	15	10 HL 11 SP

Mnemonik	Symbolische Operationen	Flags						Op.-Code			Byte	Masch.-Zyklen	Takte	Bemerkung
		C	Z	P/V	S	N	H	76	543	210				
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
SBC HL, dd	HL ← HL—dd —CY	↕	↕	V	↕	1	×	11 01	101 dd0	101 010	2	4	15	
ADD IX, pp	IX ← IX+pp	↕	.	.	.	0	×	11 00	011 pp1	101 001	2	4	15	pp Reg.-Paar 00 BC 01 DE
ADD IY, pp	IY ← IY+pp	↕	.	.	.	0	×	11 00	111 pp1	101 001	2	4	15	10 IX (IY) 11 SP
INC dd	dd ← dd+1	.	.	.	.	.	.	00	dd0	011	1	1	6	
INC IX	IX ← IX+1	.	.	.	.	.	.	11 00	011 100	101 011	2	2	10	
INC IY	IY ← IY+1	.	.	.	.	.	.	11 00	111 100	101 011	2	2	10	
DEC dd	dd ← dd—1	.	.	.	.	.	.	00	dd1	011	1	1	6	
DEC IX	IX ← IX—1	.	.	.	.	.	.	11 00	011 101	101 011	2	2	10	
DEC IY	IY ← IY—1	.	.	.	.	.	.	11 10	111 101	101 011	2	2	10	
RLCA	CY 7 ← 0 A	↕	.	.	.	0	0	00	000	111	1	1	4	Linksrotation des Akkumulatorinhaltes
RLA	CY 7 ← 0 A	↕	.	.	.	0	0	00	010	111	1	1	4	Linksrotation des Akkumulatorinhaltes durch Carry-Flag
RRCA	7 → 0 CY	↕	.	.	.	0	0	00	001	111	1	1	4	Rechtsrotation des Akkumulatorinhaltes
RRA	7 → 0 CY A	↕	.	.	.	0	0	00	011	111	1	1	4	Rechtsrotation des Akkumulatorinhaltes durch Carry-Flag
RLC r		↕	↕	P	↕	0	0	11 00	001 000	011 r	2	2	8	Linksrotation des Registers r
RLC M	CY 7 ← 0 r, M, (IX+d)	↕	↕	P	↕	0	0	11 00	001 000	011 110	2	4	15	r ZVE-Register 000 B 001 C 010 D
RLC (IX+d)		↕	↕	P	↕	0	0	11 11 ← 00	011 001 d 000	101 011 → 110	4	6	23	011 E 100 H 101 L 111 A
RLC (IY+d)	CY 7 ← 0 (IY+d)	↕	↕	P	↕	0	0	11 11 ← 00	111 001 d 000	101 011 → 110	4	6	23	
RL s	CY 7 ← 0	↕	↕	P	↕	0	0		010					Format und Zustände wie bei RLC-Befehlen,
RRC s	7 → 0 CY	↕	↕	P	↕	0	0		001					s ist entweder r, M, (IX+d) oder (IY+d)
RR s	7 → 0 CY	↕	↕	P	↕	0	0		011					analog den RLC-Befehlen Austausch von 000 im
SLA s	CY 7 ← 0 0	↕	↕	P	↕	0	0		100					Op.-Code mit den jeweils umrandeten Bits

Mnemonik	symbolische Operationen	Flags						Op.-Code			Byte	Masch. Zyklen	Takte	Bemerkung
		C	Z	P/V	S	N	H	76	543	210				
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
SRA s	7 → 0 CY	↕	↕	P	↕	0	0		101					
SRL s	0 → 7 → 0 → CY	↕	↕	P	↕	0	0		111					
RLD	A 7430 7430 M	.	↕	P	↕	0	0	11 01	101 101	101 111	2	5	18	Links- und Rechtsrotation der Tetraden des Akkumulatorinhaltes
RRD	A 7430 7430 M	.	↕	P	↕	0	0	11 01	101 100	101 111	2	5	18	und des durch HL adressierten Speicherplatzes (die obere Tetrade wird nicht mitrotiert)
BIT b, r	$Z \leftarrow /r_b$	.	↕	X	X	0	1	11 01	001 b	011 r	2	2	8	
BIT b, M	$Z \leftarrow /M_b$	.	↕	X	X	0	1	11	001	011	2	3	12	BIT setzt Z-Flag mit dem durch b ausgewählten negierten Bit des Registers r, des Speicherinhaltes M oder des Inhaltes des durch IX+d bzw. IY+d adressierten Speicherplatzes
BIT b, (IX+d)	$Z \leftarrow /(IX+d)_b$	.		X	X	0	1	11 11 ← 01	011 001 d b	101 011 → 110	4	5	20	
BIT b, (IY+d)	$Z \leftarrow /(IY+d)_b$	.	↕	X	X	0	1	11 11 ← 01	111 001 d b	101 011 → 110	4	5	20	r ZVE-Register 000 B 001 C 010 D
SET b, r	$rb \leftarrow 1$	.	.	.	.	.	.	11 11	001 b	011 r	2	2	8	011 E 100 H 101 L 111 A
SET b, M	$Mb \leftarrow 1$	.	.	.	.	.	.	11 11	001 b	011 110	2	4	15	b Bit SET setzt das 000 0 durch b aus-
SET b, (IX+d)	$(IX+d)_b \leftarrow 1$	.	.	.	.	.	.	11 11 ← 11	011 001 d b	101 011 → 110	4	6	23	001 1 gewählte Bit des 010 2 Registers r, 011 3 des Speicher- 100 4 inhalts M oder
SET b, (IY+d)	$(IY+d)_b \leftarrow 1$	.	.	.	.	.	.	11 1 ← 11	111 001 d b	101 011 → 110	4	6	23	101 5 des Inhalts des 110 6 durch IX+d bzw. 111 7 IY+d adressierten Speicherplatzes auf 1
RES b, s	$s_b \leftarrow 0$ s = r, M, (IX+d), (IY+d)	.	.	.	.	.	.	10						Flags und Takte analog den SET-Befehlen Austausch von 11 im Op.-Code mit 10
JMP nn	PC ← nn	.	.	.	.	.	.	11 ← ←	000 n n	011 → →	3	3	10	Unbedingter Sprung
JPNZ nn	PC ← nn, falls Beding. erf., sonst nächster Bef.	.	.	.	.	.	.	11 ← ←	000 n n	010 → →	3	3	10	Sprung, wenn Z-Flag = 0

Mnemonik	Symbolische Operationen	Flags						Op.-Code			Byte	Maschinen-Zyklen	Takte	Bemerkung
		C	Z	P/V	S	N	H	76	543	210				
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
JPZ nn	PC ← nn, falls Beding. erf., sonst nächster Bef.	.	.	.	.	.	.	11 ← ←	001 n n	010 → →	3	3	10	Sprung, wenn Z-Flag = 1
JPNC nn	PC ← nn, falls Beding. erf., sonst nächster Bef.	.	.	.	.	.	.	11 ← ←	010 n n	010 → →	3	3	10	Sprung, wenn C-Flag = 0
JPC nn	PC ← nn, falls Beding. erf., sonst nächster Bef.	.	.	.	.	.	.	11 ← ←	011 n n	010 → →	3	3	10	Sprung, wenn C-Flag = 1
JPPO nn	PC ← nn, falls Beding. erf., sonst nächster Bef.	.	.	.	.	.	.	11 ← ←	100 n n	010 → →	3	3	10	Sprung, wenn P-Flag = 0
JPPE nn	PC ← nn, falls. Beding. erf., sonst nächster Bef.	.	.	.	.	.	.	11 ← ←	101 n n	010 → →	3	3	10	Sprung, wenn P-Flag = 1
JPP nn	PC ← nn, falls Beding. erf., sonst nächster Bef.	.	.	.	.	.	.	11 ← ←	110 n n	010 → →	3	3	10	Sprung, wenn S-Flag = 0
JPM nn	PC ← nn, falls Beding. erf., sonst nächster Bef.	.	.	.	.	.	.	11 ← ←	111 n n	010 → →	3	3	10	Sprung, wenn S-Flag = 1
JR e	PC ← PC+e	.	.	.	.	.	.	00 ←	011 e—2	000 →	2	3	12	Sprung zu der um die Verschiebung e größeren Speicheradresse
JRC e	PC ← PC+e, falls Beding. erf., sonst nächster Bef.	.	.	.	.	.	.	00	111	000	2	3	12	Sprung, wenn C-Flag = 1
											2	2	7	sonst
JRNC e	PC ← PC+e, falls Beding. erf., sonst nächster Bef.	.	.	.	.	.	.	00 ←	110 e—2	000 →	2	3	12	
											2	2	7	sonst
JRZ e	PC ← PC+e, falls Beding. erf., sonst nächster Bef.	.	.	.	.	.	.	00	101	000	2	3	12	Sprung, wenn Z-Flag = 1
											2	2	7	sonst
JRNZ e	PC ← PC+e, falls Beding. erf. sonst nächster Bef.	.	.	.	.	.	.	00 ←	100 e—2	000 →	2	3	12	Sprung, wenn Z-Flag = 0
											2	2	7	sonst
JMP M	PC ← M	.	.	.	.	.	.	11	101	001	1	1	4	
JMP (IX)	PC ← IX	.	.	.	.	.	.	11 11	011 101	101 001	2	2	8	
JMP (IY)	PC ← IY	.	.	.	.	.	.	11 11	111 101	101 001	2	2	8	

Mnemonik	Symbolische Operationen	Flags						Op.-Code			Byte	Masch.-Zyklen	Takte	Bemerkung
		C	Z	P/V	S	N	H	76	543	510				
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
DJNZ e	B ← B—1 PC ← PC+e, falls B ≠ 0, falls B = 0 nächster Bef.	.	.	.	.	.	.	00 ←	010 e—2	000 →	2 2	3 2	13 8	Sprung, wenn B ≠ 0 wenn B = 0
CALL nn	(SP—1) ← PC_H (SP—2) ← PC_L PC → nn	.	.	.	.	.	.	11 ← ←	001 n n	101 → →	3	5	17	
CANZ nn	(SP—1) ← PC_H (SP—2) ← PC_L PC → nn, falls Beding. erf. sonst nächster Bef.	.	.	.	.	.	.	11 ←	000 n	100 →	3	5	17	**bedingte UP-Aufrufe** Falls die Bedingung erfüllt ist, wird der augenblickliche PC im Stapel zwischengespeichert und das Programm ab Adresse nn fortgesetzt
CAZ nn	(SP—1) ← PC_H (SP—2) ← PC_L PC ← nn, falls Beding. erf. sonst nächster Bef.	.	.	.	.	.	.	11 ← ←	001 n n	100 → →	3	5	17	Bei nicht erfüllter Bedingung erfolgt Abarbeitung des nächsten Befehls
CANC nn	(SP—1) ← PC_H (SP—2) ← PC_L PC ← nn, falls Beding. erf. sonst nächster Bef.	.	.	.	.	.	.	11 ← ←	010 n n	100 → →	3 3	5 3	17 10	Für UP-Aufruf gilt bei Angabe von Byte, Masch.-Zyklen, Takten die erste Zeile, sonst die zweite Zeile
CAC nn	(SP—1) ← PC_H (SP—2) ← PC_L PC ← nn, falls Beding. erf. sonst nächster Bef.	.	.	.	.	.	.	11 ← ←	011 n n	000 → →	3 3	5 3	17 10	
CAPO nn	(SP—1) ← PC_H (SP—2) ← PC_L PC ← nn, falls Beding. erf. sonst nächster Bef.	.	.	.	.	.	.	11 ←	100 n	100 →	3 3	5 3	17 10	
CAPE nn	(SP—1) ← PC_H (SP—2) ← PC_L PC ← nn, falls Beding. erf. sonst nächster Bef.	.	.	.	.	.	.	11 ← ←	101 n n	100 → →	3 3	5 5	17 10	
CAP nn	siehe andere bedingte UP-Aufrufe	.	.	.	.	.	.	11 ← ←	110 n n	100 → →	3	5	17	
CAM nn	siehe andere bedingte UP-Aufrufe	.	.	.	.	.	.	11 ← ←	111 n n	100 → →	3 3	5 3	17 10	
RET	PC_L ← (SP) PC_H ← (SP+1)	.	.	.	.	.	.	11	001	001	1	3	10	Unbedingte Rückkehr zur Aufrufstelle

Mnemonik	Symbolische Operationen	Flags						Op.-Code			Byte	Masch.-Zyklen	Takte	Bemerkung
		C	Z	P/V	S	N	H	76	543	210				
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RNZ	$PC_L \leftarrow (SP)$ $PC_H \leftarrow (SP+1)$ falls Beding. erf., sonst nächster Bef.	.	.	.	.	.	.	11	000	000	1 1	3 1	11 5	Falls die Bedingung erfüllt ist, wird der im Stapel zwischengespeicherte PC in den PC zurückgeladen und das Programm kehrt so an die Aufrufstelle zurück
RZ	$PC_L \rightarrow (SP)$ $PC_H \rightarrow (SP+1)$ falls Beding. erf., sonst nächsten Bef.	.	.	.	.	.	.	11	001	000	1 1	3 1	11 5	Bei nicht erfüllter Bedingung erfolgt Abarbeitung des nächsten Befehls
RNC	$PC_L \rightarrow (SP)$ $PC_H \rightarrow (SP+1)$ falls Beding. erf., sonst nächster Bef.	.	.	.	.	.	.	11	010	000	1 1	3 1	11 5	Für die erfüllte Bedingung gilt bei Angabe von Byte, Masch.-Zyklen, Takten die erste Zeile, sonst die zweite Zeile
RC	$PC_L \leftarrow (SP)$ $PC_H \leftarrow (SP+1)$ falls Beding. erf., sonst nächster Bef.	.	.	.	.	.	.	11	011	000	1 1	3 1	11 5	
RPO	$PC_L \leftarrow (SP)$ $(PC_H \leftarrow (SP+1)$ falls Beding. erf., sonst nächster Bef.	.	.	.	.	.	.	11	100	000	1 1	3 1	11 5	
RPE	$PC_L \leftarrow (SP)$ $PC_H \leftarrow (SP+1)$ falls Beding. erf., sonst nächster Bef.	.	.	.	.	.	.	11	101	000	1 1	3 1	11 5	
RP	$PC_L \leftarrow (SP)$ $PC_H \leftarrow (SP+1)$ falls Beding. erf., sonst nächster Bef.	.	.	.	.	.	.	11	110	000	1 1	3 1	11 5	
RM	$PC_L \leftarrow (SP)$ $PC_H \leftarrow (SP+1)$ falls Beding. erf., sonst nächster Bef.	.	.	.	.	.	.	11	111	000	1 1	3 1	11 5	
RETI	Rückkehr vom Interrupt	.	.	.	.	.	.	11 01	101 001	101 101	2	4	14	
RETN	Rückkehr vom nichtmaskierten Interrupt	.	.	.	.	.	.	11 01	101 000	101 101	2	4	14	
RST p	$(SP-1) \leftarrow PCH$ $(SP-2) \leftarrow PCL$ $PCH \leftarrow 0$ $PCL \leftarrow P$	.	.	.	.	.	.	11	f	111	1	3	11	f p 000 00H 001 08H 010 10H 011 18H 100 20H 101 28H 110 30H 111 38H

Mnemonik	Symbolische Operationen	Flags						Op.-Code			Byte	Masch.-Zyklen	Takte	Bemerkung
		C	Z	P/V	S	N	H	76	543	210				
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
IN n	A ← (n)	.	.	.	.	.	.	11 ←	011 n	011 →	2	3	10	Für alle I/O-Befehle gilt: Toradresse über A0 · · · A7 des Adreßbuses
IN r	r ← (C)	.	↕	F	↕	0	↕	11 01	101 r	101 000	2	3	11	Daten vom (Input) zum (Output) ausgewählten Tor über A8 · · · A15 des Adreß-Bus
INF	Setzt nur Flags	.	↕	P	↕	0	↕	11 01	101 110	101 000	2	3	11	
INI	M ← (C) B ← B—1 HL ← (HL+1)	.	↕ ③	X	X	1	X	11 10	101 100	101 010	2	4	15	
INIR	M ← (C) B ← B—1 HL ← HL+1 wiederholen, bis B = 0	.	1 ③	X	X	1	X	11 10	101 110	101 010	2 2	5 4	10 15	wenn B ≠ 0 wenn B = 0
IND	M ← (C) B ← B—1 HL ← HL—1	.	↕	X	X	1	X	11 10	101 101	101 010	2	4	15	
INDR	M ← (C) B ← B—1 HL ← HL—1 wiederholen, bis B = 0	.	1	X	X	1	X	11 10	101 111	101 010	2 2	5 4	20 15	wenn B ≠ 0 wenn B = 0
OUT n	(n) ← A	.	.	.	.	.	.	11 →	010 n	011 →	2	3	11	
OUT r	(C) ← r	.	. ③	.	.	.	.	11 01	101 r	101 001	2	3	12	
OUTI	(C) ← M B ← B—1 HL ← HL+1	.	↕	X	X	1	X	11 10	101 100	101 011	2	4	15	
OUTIR	(C) ← M B ← B—1 HL ← HL+1 wiederholen, bis B = 0	.	1 ③	X	X	1	X	11 10	101 110	101 011	2 2	5 4	20 15	wenn B ≠ 0 wenn B = 0
OUTD	(C) ← M B ← B—1 HL ← HL—1	.	↕	X	X	1	X	11 10	101 101	101 011	2	4	15	
OTDR	(C) ← M B ← B—1 HL ← HL—1 wiederholen, bis B = 0	.	1	X	X	1	X	11 10	101 111	101 011	2 2	5 4	20 15	wenn B ≠ 0 wenn B = 0

Zugelassene Operanden

Operanden	Bemerkungen
nn kann sein 1. ± ADR 2. ± ADR+ADR 3. ADR — ADR n kann sein: 1. ± Byte 2. ± Byte ± Byte e kann sein: d kann sein:	 —126≤e≤129 —128≤d≥127

Erläuterungen

ADR kann sein:

1. 0≤ADR≤65535 Dezimalwert
2. 0Q≤ADR≤177777Q Oktalwert, Kennzeichnung mit Q oder O
3. 0H≤ADR≤0FFFFH Hexadezimalwert, 1. Zeichen muß eine Ziffer sein, gekennzeichnet zum Abschluß mit H
4. 0B≤ADR≤ 1111111111111111B Binärwert, Kennzeichnung mit B
5. Symbol XXXXX, externes Symbol YY.XXXXX (PN und Symbol)
6. Befehlszähler
   Byte kann sein:
   1. Dezimalwert
      Ø≤Byte≤255
   2. Oktalwert
      ØQ≤Byte≤377Q
   3. Hexadezimalwert
      ØH≤Byte≤ØFFH
   4. Binärwert
      ØB≥Byte≤11111111B
   5. Untere 8 Bit eines 16- Bit-Operanden L (ADR)
   6. Obere 8 Bit eines 16-Bt-iOperanden H (ADR)
   7. Symbol
   8. Externes Symbol
   9. Literal 'X' X ist ein Zeichen aus dem Zeichenvorrat nach TGL 23 207/1

Mnemonik	Operand	Operationsbeschreibung
PN	XX	Programmname (2 Buchstaben gewertet, Rest Kommentar)
ORG	Absolute Adresse nn	Adressiert den absoluten Programmanfang $0 \leq X \leq 64$ K (darf nur 1mal am Programmanfang stehen)
EQU	n oder nn	Definiert Symbol A einmalig
DB	n oder 'XX···X'	Definiert 1-Byte-Konstante Zeichenkette (ISO-Code)
DA	nn	Definiert 2-Byte-Konstante in der Reihenfolge L(nn), H(nn)
TITL	'XX···X'	Anweisung zum Druck der Überschrift
EJEC	—	Anweisung zur Seitenschaltung
MACR	Parameterliste	Makrodefinition als Erweiterung vorgesehen, aber noch nicht festgelegt
ENDM	—	
IF	nn oder n	Bedingte Assemblierung
ENIF	—	
BER	nn	Reserviert nn Speicherplätze
DEF	nn oder n	Definiert Symbol bis zur nächsten Definition

robotron

VEB Robotron-Elektronik
Zella-Mehlis
DDR - 606 Zella-Mehlis
Straße der Antifa 63-66

Exporteur:

Robotron Export-Import
Volkseigener
Außenhandelsbetrieb
der Deutschen Demokratischen
Republik
DDR - 108 Berlin
Friedrichstraße 61

DEWAG DRESDEN